

Oracle® Cloud at Customer

Getting Started with Oracle Cloud at Customer



Release 18.1.4

E88605-17

June 2021

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Cloud at Customer Getting Started with Oracle Cloud at Customer, Release 18.1.4

E88605-17

Copyright © 2017, 2021, Oracle and/or its affiliates.

Primary Authors: Shynitha Shanthakumar, Peter LaQuerre, Kumar Dhanagopal

Contributing Authors: John Bigane, Salvador Esparza, Albert Leigh, Karen Orozco, Gavin Parish, Eric Lyke

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	vii
Related Documents	vii
Conventions	vii

Part I Signing In and Getting Started with Oracle Cloud at Customer

1 Introduction to Oracle Cloud at Customer

About Oracle Cloud at Customer	1-1
Components of Oracle Cloud at Customer	1-1
Services Available on Oracle Cloud at Customer	1-2
About Oracle Cloud at Customer Subscriptions	1-2
About Hardware Subscription	1-2
About Software Subscription	1-2
About Universal Credits on Cloud at Customer	1-3
About Your Cloud at Customer Data Region	1-4

2 Oracle Cloud at Customer Responsibilities

About Preparing for an Oracle Cloud at Customer Delivery	2-1
Initial Configuration and Setup	2-1
Day-to-day Systems Management	2-2
About Managing Your Oracle Cloud Account	2-2

3 About Your Cloud at Customer IaaS Account

What is Your Cloud at Customer IaaS Account	3-1
Sign in to Your IaaS Account	3-1

4	Sign In to Your Cloud at Customer Account	
	Roadmap for Signing In to Your Cloud at Customer Account	4-1
	Sign In to Your Cloud Account on Oracle Cloud	4-3
	Activate Your Cloud Account	4-4
	Sign In to Your Cloud Account For the First Time	4-4
	Extend Your Cloud Account to Your Cloud at Customer Region	4-5
	Sign In to Your Cloud at Customer Data Region	4-6
5	Get Started with Your Cloud at Customer Account	
	Explore the My Services Dashboard	5-1
	Basic Features of the My Services Dashboard	5-1
	The Welcome Section	5-2
	The Cloud Services Section	5-3
	View Oracle Cloud Service Details	5-6
	Access a Cloud Service Console	5-7
	Create Instances	5-7
	About the Documentation for Oracle Cloud Services	5-8
6	Monitor Your Usage and Universal Credits Balance for Oracle Cloud at Customer	
	Sign In to Your Oracle Cloud Account to Check Your Balance	6-1
	Check Your Account Balance and Usage Summary	6-2
	Download Your Account Balance and Usage Summary	6-2
	Obtain Usage Data for Your Cloud at Customer Region	6-3
	Set an Alert to Monitor Your Account Balance	6-3
	Use the REST API to Check Your Account Balance	6-5
7	Monitor Your Cloud Service Performance	
	View Performance Metrics for an Oracle Cloud Service	7-1
	Set an Alert for a Performance Metric	7-2
8	Create and Manage Users for Oracle Cloud at Customer	
	Sign In to Your Oracle Cloud at Customer Account to Create Users	8-1
	About the Users Page in a Cloud Account	8-1
	Create a New Cloud Account User	8-2
	Create Groups	8-2
	Assign Cloud Account Roles to a User	8-3

Part II Getting Started with IaaS and PaaS Services

9 Compute Classic: Basic Tasks

Create an Oracle Cloud User with the Required Roles	9-1
Generate an SSH Key Pair	9-2
Create an Oracle Linux Instance	9-4
Create an Oracle Linux Instance Using a Nonpersistent Boot Disk	9-5
View Details of an Instance	9-6
Enable SSH Access to a VM	9-7
Log In to a VM Using SSH	9-8
Add an SSH-Enabled User	9-9
Reboot an Instance	9-10
Shut Down and Restart an Instance	9-10
Monitor Metrics for Your VMs	9-11
Change the Shape of an Instance	9-12
Create a Storage Volume	9-13
Attach a Volume to a VM	9-13
Mount a Volume	9-14
Retrieve Predefined Instance Metadata	9-15
Delete and Re-create an Instance	9-17

10 Compute Classic: Advanced Tasks

Control Network Traffic	10-1
Create a Bootable Volume	10-3
Create an Instance Snapshot	10-4
Register a Machine Image	10-5
Create a Colocated Volume Snapshot	10-5
Restore a Volume from a Snapshot	10-6
Create Resources Using an Orchestration	10-6
Create a Multi-Tier Topology with IP Networks Using an Orchestration	10-10
Manage Resources Using Terraform	10-22
Create a Multi-Tier Topology with IP Networks Using Terraform	10-33

11 Compute Classic: Using the REST API

Prepare to Use the REST API	11-1
Get an Authentication Token	11-2

Get the Details of a VM Using the REST API	11-4
Add Block Storage for a VM Using the REST API	11-4

12 Object Storage Classic: Managing Containers and Objects

Get an Authentication Token	12-1
Create a Container	12-2
List the Containers in the Account	12-3
Upload a Large File	12-3
Download a File	12-8
Copy an Object	12-8
List the Objects in a Container	12-9
Delete an Object	12-9
Delete a Container	12-10

A Additional Cloud at Customer Tasks

Web Browser Requirements	A-1
Change Your Cloud Account Password	A-1

Preface

Getting Started with Oracle Cloud at Customer introduces you to the roles and responsibilities Oracle Operations team, as well your customer responsibilities, when you purchase Oracle Cloud at Customer. It also introduces you to managing your Oracle Cloud Account on Oracle Cloud at Customer, using the My Services Dashboard.

Topics

- [Audience](#)
- [Related Documents](#)
- [Conventions](#)

Audience

This document is primarily for Oracle customers responsible for managing the Oracle Cloud Account and the Oracle Cloud Services available on Oracle Cloud at Customer. As part of managing the account, these administrators can add additional users and create Oracle Cloud service instances on Oracle Cloud at Customer.

Related Documents

For more information, see these Oracle resources:

- <http://cloud.oracle.com>
- *What's New for Oracle Cloud at Customer*
- *Oracle Cloud at Customer Deployment Guide*
- *Getting Started with Oracle Cloud*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Part I

Signing In and Getting Started with Oracle Cloud at Customer

This part describes how to sign in to Oracle Cloud at Customer and get started with it.

Topics

- [Introduction to Oracle Cloud at Customer](#)
- [Oracle Cloud at Customer Responsibilities](#)
- [About Your Cloud at Customer IaaS Account](#)
- [Sign In to Your Cloud at Customer Account](#)
- [Get Started with Your Cloud at Customer Account](#)
- [Monitor Your Usage and Universal Credits Balance for Oracle Cloud at Customer](#)
- [Monitor Your Cloud Service Performance](#)
- [Create and Manage Users for Oracle Cloud at Customer](#)

1

Introduction to Oracle Cloud at Customer

Oracle Cloud at Customer provides the power and efficiency of Oracle Cloud in your data center.

Topics

- [About Oracle Cloud at Customer](#)
- [Components of Oracle Cloud at Customer](#)
- [Services Available on Oracle Cloud at Customer](#)
- [About Oracle Cloud at Customer Subscriptions](#)
- [About Universal Credits on Cloud at Customer](#)
- [About Your Cloud at Customer Data Region](#)

About Oracle Cloud at Customer

Oracle Cloud at Customer is a family of products that deliver all the benefits of Oracle Cloud to your data center.

When you subscribe to Oracle Cloud at Customer, you subscribe to the hardware and software required to use Oracle Cloud services in your own data center. Oracle installs, configures, monitors, and manages the Oracle Cloud at Customer environment for you, while you take full advantage of the control, security, and networking features of your data center.

For more information, see [Cloud at Customer](#) on the Oracle Cloud Web site.

Components of Oracle Cloud at Customer

Oracle Cloud at Customer consists of several important components that together provide you with the Oracle Cloud experience in your data center.

Each Oracle Cloud at Customer subscription includes:

- **The hardware required to run Oracle Cloud at Customer**, which is installed and set up by Oracle Field Service Engineers in your data center. This might include one or more Cloud at Customer racks.
- **The Oracle Cloud at Customer control plane software**, which consists of the software infrastructure required to run Oracle Cloud. The control plane is installed, configured, and managed by Oracle. It is modeled after and nearly identical to the control plane software used to run Oracle Cloud software hosted in the Oracle data centers throughout the world.
- **The Oracle Advanced Support Gateway (OASG)**, which includes the hardware and software required to securely connect your Oracle Cloud at Customer environment to the remote Oracle Operations team. The remote Operations team manages the day-to-day operations of Oracle Cloud at Customer, just as it manages Oracle Cloud hosted in Oracle data centers.

- **The Oracle Cloud Services** to which you have subscribed. This can include many of the Infrastructure as a Service (IaaS) services, such as Oracle Cloud Infrastructure Compute Classic, and Platform as a Service (PaaS) offerings, such as Oracle Java Cloud Service and Oracle Database Cloud Service.

Services Available on Oracle Cloud at Customer

For the list of supported Cloud at Customer services along with version details, see [What's New for Oracle Cloud at Customer](#). For Object Storage Classic, Java Cloud Service, and Database Cloud Service on Cloud at Customer, see [Getting Started with IaaS and PaaS Services](#).

About Oracle Cloud at Customer Subscriptions

A subscription to Oracle Cloud at Customer consists of both a hardware subscription and a software subscription. Both are required to run Oracle Cloud in your data center.

Review the following topics to know more about the subscription types:

- [About Hardware Subscription](#)
- [About Software Subscription](#)

About Hardware Subscription

The Cloud at Customer hardware subscription consists of:

- The hardware required to run the Oracle Cloud control plane, which is the base software stack required to run Oracle Cloud.
- The optional hardware required for additional compute nodes, object storage, or block storage.
- Optional hardware required if you are subscribing to Exadata Cloud at Customer.

About Software Subscription

The Oracle Cloud at Customer software subscription is similar to Oracle Cloud. Starting with Oracle Cloud at Customer 18.1.4, all new Oracle Cloud at Customer subscriptions support Universal Credits.

The subscription type available for Oracle Cloud at Customer is ExaCC (Control Plane only), which comes with Universal Credits. To sign in to the Oracle Cloud at Customer account with ExaCC (Control Plane only), use the Welcome email. See [Table 4-1](#).

For information about Universal credits, see [About Universal Credits on Cloud at Customer](#).

For sign-in procedures, see [Roadmap for Signing In to Your Cloud at Customer Account](#).

About Universal Credits on Cloud at Customer

Universal Credits is the next-generation subscription model offered by Oracle Cloud. Universal Credits offers some key benefits that can improve your Oracle Cloud experience.

It provides:

- A simpler buying experience
 - You can subscribe to a single set of credits (for all PaaS services).
 - You are not locked into using specific service or subscription SKU.
- Greater flexibility
 - You have access to new services as soon as they are introduced. No need to contact Oracle Sales or modify your order.
 - Your Universal Credits can be used for any new Platform as a Service services you use.
 - You can sign up for a Pay As You Go subscription with no up-front costs or you save money by committing to a Monthly Flex subscription.
- Easier expansion process
 - You can elastically scale with confidence with benefits of lower pre-paid instance pricing.
 - You can expand to new workloads without going through another contracting cycle.
- A single set of cloud credits that spans Cloud at Customer and Oracle Cloud
 - Available with Oracle Cloud at Customer 18.1.4. Contact Oracle Sales for details and restrictions.

The Universal Credits subscription model provides a flexible buying and usage model for Oracle Cloud Services. With this subscription model, all customers of Oracle Cloud at Customer who have signed up for the Universal Credits subscription model have access to all eligible PaaS services.

Note that your Universal Credits subscription extends to PaaS services only on Cloud at Customer. IaaS services on Cloud at Customer are provisioned as part of a non-metered cloud service subscription. For more information, contact your Oracle Sales representative.

If you are an existing Cloud at Customer user, and you purchase a new Cloud at Customer Universal Credits subscription, you will be provisioned with a new, additional Cloud Account for the services that support Universal Credits. Your new Universal Credits account will have its own URL and sign-in credentials.

There are two payment options:

- Sign up for a **Pay As You Go** subscription to use Oracle Cloud with no up-front costs. Oracle charges you in arrears for your amount of usage each month.

 **Note:**

This option is not supported on Oracle Cloud at Customer for Exadata Cloud at Customer.

- Sign up for a **Monthly Flex** subscription to commit to a specific monthly payment, based on your estimated monthly usage. Use the cost estimator to help you estimate your monthly usage. To qualify for a monthly flex subscription, you commit to a minimum of US\$ 1000 a month for at least one year, and you are charged for your Oracle Cloud usage at a lower rate than the Pay As You Go subscription. The lower rate is calculated, based on your monthly usage commitment and your contract duration.

For information, see [Oracle Universal Credit Pricing](#).

About Your Cloud at Customer Data Region

Oracle data centers are grouped into data regions throughout the world. When you sign up for an Oracle Cloud account, you select the default data region for your Oracle Cloud subscription. You can also extend your subscription to another Oracle data region. When you sign up for Oracle Cloud at Customer, you can access Cloud at Customer just as you access other data regions throughout the world.

Why Are Multiple Data Regions Useful?

Extending your subscription to another data region is useful if you want to access cloud services that are not available in your default data region or if you have employees stationed in other regions of the world. When you extend your subscription to another data region, you are granted a separate set of sign-in credentials for the additional data region, but you can use your existing Universal Credits across both regions.

About the Oracle Cloud at Customer Data Region

Your Cloud at Customer installation (in your data center) is treated as an additional data region available only to your Oracle Cloud subscription. When you subscribe to Oracle Cloud at Customer with Universal Credits, you are first granted access to an Oracle Cloud Account with Universal Credits (in an Oracle data region). You can then extend your Oracle Cloud subscription to access your Oracle Cloud at Customer data region.

Benefits of Accessing Cloud at Customer as a Data Region

When you extend your Cloud account on Oracle Cloud to your Oracle Cloud at Customer data region, you can access services both on Oracle Cloud (in an Oracle data center) and on Oracle Cloud at Customer (in your data center). You can also share your Universal Credits across your Oracle Cloud and Oracle Cloud at Customer services.

How Do I Switch Between My Oracle Cloud and Cloud at Customer Data Regions?

Like other data regions, your Oracle Cloud at Customer data region is accessed via a separate set of sign-in credentials, and you can use your Universal Credits for the services and resources you use in your Oracle Cloud at Customer data region.

2

Oracle Cloud at Customer Responsibilities

When you subscribe to Oracle Cloud at Customer, you should understand the responsibilities of the Oracle Operations and Support personnel, as well as your responsibilities as a customer.

Topics

- [About Preparing for an Oracle Cloud at Customer Delivery](#)
- [Initial Configuration and Setup](#)
- [Day-to-day Systems Management](#)
- [About Managing Your Oracle Cloud Account](#)

About Preparing for an Oracle Cloud at Customer Delivery

One of your key responsibilities as an Oracle Cloud at Customer customer is to work with your assigned Oracle Field Engineer or Oracle Advanced Customer Support engineer to ensure your data center meets all the requirements of the Oracle Cloud at Customer hardware.

This process includes your participation in an audit of your data center by your Oracle Support representative. The Oracle representative will make sure there are no issues that will delay or prevent the Oracle Cloud at Customer from being installed and configured quickly, efficiently, and securely in your data center.

To help with this process, be sure to review the *Oracle Cloud at Customer Deployment Guide*, which lists the data center requirements for Oracle Cloud at Customer.

Initial Configuration and Setup

When your Oracle Cloud at Customer hardware arrives, a team of Oracle customer support engineers will set up the hardware, install and configure the Oracle Cloud control plane, and install and configure the Oracle Advanced Support Gateway.

More specifically:

- An Oracle Field Service engineer will set up and configure the hardware.
- An Oracle Advanced Customer Support (ACS) engineer will install and configure the software.
- A member of the Oracle Gateway Team will set up and configure the Oracle Advanced Support Gateway.

When the machine is up and running, the ACS engineer validates the installation and completes your Oracle Cloud at Customer order. The designated Oracle Cloud Account administrator on your team receives a welcome email message. The email contains the links and credentials required to log in to your new Cloud Account for the first time.

Day-to-day Systems Management

Day-to-day management of your Oracle Cloud at Customer environment, including the Oracle hardware and software, is handled by Oracle Operations, via the Advanced Support Gateway.

The Oracle Advanced Support Gateway is also managed by Oracle Operations. It provides efficient, secure connections between your Oracle Cloud at Customer hardware and software and the Oracle Operations team. Using the gateway, the Oracle Cloud Operations team monitors your system and responds to your service requests securely and promptly.

The goal is to free up your Information Technology (IT) engineers so they can support the real work that your company needs to perform, including developing and deploying applications, managing the Oracle Cloud services, and running your business.

About Managing Your Oracle Cloud Account

Oracle Cloud Account management is the responsibility of you, the customer, and you can assign administrators to manage the account and your subscribed services, just as you do on Oracle Public Cloud.

For example, the initial, designated Oracle Cloud Account Administrator on your team is responsible for receiving the Welcome email and initially logging in to your Oracle Cloud Account on Oracle Cloud at Customer. The initial account administrator can then create additional administrator accounts.

Typically, two or more Oracle Cloud Account administrators on your team to manage your Oracle Cloud Account. They monitor your Cloud Account usage and create and manage Cloud Service instances. They also can create additional Cloud users, who can be assigned specific tasks or roles within the account.

See [Sign In to Your Cloud at Customer Account](#) and [Create and Manage Users for Oracle Cloud at Customer](#).

3

About Your Cloud at Customer IaaS Account

When your Oracle Cloud at Customer environment is initially installed and configured, you are provided an Oracle Cloud account, which provides you with basic Infrastructure as a Service (IaaS) services, running on your Oracle Cloud at Customer hardware.

Topics

- [What is Your Cloud at Customer IaaS Account](#)
- [Sign in to Your IaaS Account](#)

What is Your Cloud at Customer IaaS Account

When the Oracle Cloud at Customer rack is registered with Oracle Cloud, an initial Infrastructure as a Service (IaaS)-only Cloud Account is provisioned on the Oracle Cloud at Customer rack.

With this account, you can access the following IaaS services:

- Compute Classic
- Storage Classic
- Load Balancing Classic

The Oracle Cloud at Customer IaaS Account is a default Cloud Account to make sure you have access to the Cloud at Customer rack. When you sign up for Universal Credits, you are provisioned with a separate Oracle Cloud Account with Universal Credits, which allows you to use your Universal Credits in an Oracle data region and in your Oracle Cloud at Customer data region.

Sign in to Your IaaS Account

A set of emails are generated when the Oracle Cloud at Customer is registered with Oracle Cloud. You can use these emails to activate and access your IaaS account.

The first set of emails provide the information you need to activate and then sign-in to your Oracle Cloud at Customer IaaS Account. You can use this account to verify and get started with IaaS services on your Oracle Cloud at Customer hardware. You are charged for these services using a standard metered or non-metered subscription model and not the Universal Credits subscription model.

Note that when you receive your activation email, you should work with your Oracle representatives to be sure the system is ready to use.

If you are subscribing to an Oracle Cloud Account with Universal Credits, you might not need to access your Cloud at Customer IaaS account. Instead, you will later receive separate emails that provide information about activating and signing in to your Oracle Cloud Account with Universal Credits.

4

Sign In to Your Cloud at Customer Account

Sign in to your Oracle Cloud account with Universal Credits, extend your Cloud account to Cloud at Customer data region, and then sign in to your Cloud at Customer account with Universal Credits.



Note:

Review the [Web Browser Requirements](#) and ensure that you use a supported web browser to perform the tasks in this guide.

Topics

- [Roadmap for Signing In to Your Cloud at Customer Account](#)
- [Sign In to Your Cloud Account on Oracle Cloud](#)
- [Extend Your Cloud Account to Your Cloud at Customer Region](#)
- [Sign In to Your Cloud at Customer Data Region](#)

Roadmap for Signing In to Your Cloud at Customer Account

This topic describes the steps for signing in to Oracle Cloud at Customer account with ExaCC (Control Plane only).

Table 4-1 Steps for Signing In to the Cloud at Customer Account

Step #	Step	Description	Performed by	More Information
1	Place the initial order with Oracle Sales team.	When you place an order for Oracle Cloud at Customer, a new Oracle Cloud account is provisioned and set to Pending Registration. After the initial order, the Oracle Cloud at Customer hardware is delivered to your data center, and the Oracle Field Service and Advanced Customer Services teams install and configure the system.	Oracle	About Preparing for an Oracle Cloud at Customer Delivery

Table 4-1 (Cont.) Steps for Signing In to the Cloud at Customer Account

Step #	Step	Description	Performed by	More Information
2	Register the Cloud at Customer with Oracle Cloud and provision the Cloud at Customer IaaS account.	<p>When the Cloud at Customer rack is registered with Oracle Cloud, an initial Infrastructure as a Service (IaaS)-only Cloud Account is provisioned on the Cloud at Customer rack.</p> <p>With this account, you can access the IaaS services like Compute Classic, Storage Classic, and Load Balancing Classic.</p>	Oracle	Initial Configuration and Setup
3	Sign in your IaaS cloud account on the Cloud at Customer rack.	<p>Use the activation email to activate your Cloud at Customer IaaS account.</p> <p>Once you activate the IaaS account, you will receive a Welcome email with the Cloud at Customer IaaS account credentials.</p> <p>Sign in to your Cloud at Customer IaaS account using the credentials in the email.</p>	Customer	About Your Cloud at Customer IaaS Account
4	Release Universal Credits order and provision the Oracle Cloud account	<p>Oracle representative releases the Universal Credits order.</p> <p>Oracle Cloud account is provisioned with Universal Credits.</p>	Oracle	
5	Sign in to your Oracle Cloud account with Universal Credits.	<p>Use the activation email to activate your Oracle Cloud account with Universal Credits.</p> <p>Once you activate the Oracle Cloud account, you will receive a Welcome email with the Oracle Cloud account credentials.</p> <p>Sign in to your Oracle Cloud account using the credentials in the email.</p>	Customer	Sign In to Your Cloud Account on Oracle Cloud
6	Extend your Oracle Cloud account to your Cloud at Customer data region and provision your Cloud at Customer account.	<p>Extend your Oracle Cloud account with Universal Credits, to your Cloud at Customer data region.</p> <p>Oracle Cloud at Customer account is provisioned with Universal Credits.</p>	Customer	Extend Your Cloud Account to Your Cloud at Customer Region

Table 4-1 (Cont.) Steps for Signing In to the Cloud at Customer Account

Step #	Description	Performed by	More Information
7	<p>Sign in to your Oracle Cloud at Customer account with Universal Credits.</p> <p>Use the activation email to activate your Oracle Cloud at Customer account with Universal Credits.</p> <p>Once you activate the Oracle Cloud at Customer account, you will receive a Welcome email with the Oracle Cloud at Customer account credentials.</p> <p>Sign in to your Oracle Cloud at Customer account using the credentials in the email.</p>	Customer	Sign In to Your Cloud at Customer Data Region

Sign In to Your Cloud Account on Oracle Cloud

Your first step in accessing your Oracle Cloud at Customer services with Universal Credits is to sign in to your new Oracle Cloud Account in your default Oracle data region.



Note:

Review the [Web Browser Requirements](#) and ensure that you use a supported web browser to perform the tasks in this guide.

When you subscribe to the Universal Credits subscription model, two emails are generated when your Cloud Account with Universal Credits are provisioned:

1. An activation email:
Use this email to activate the Oracle Cloud account with Universal Credits. Click **Activate** and fill in a short form to activate the account.
2. A welcome email:
Use this email to sign in to your Oracle Cloud account with Universal Credits. Click **Get Started with Oracle Cloud**. Enter the user name and the temporary password from the welcome email, and click **Sign In**.

You will be prompted to change your password the first time you sign in. After you sign in and change the default password, you are directed to the Guided Journey, just like any other new Oracle Cloud user.



Note:

If you need to change your password again later, see [Change Your Cloud Account Password](#).

For more information, refer to the following topics:

- [Activate Your Cloud Account](#)
- [Sign In to Your Cloud Account For the First Time](#)

Activate Your Cloud Account

When your Oracle Cloud Account with Universal Credits is provisioned, you'll receive an activation email. To activate your services, you must provide your details and set up your account with us.

Review the instructions in the email to create an account and start using your services.

1. Open the email you received from Oracle Cloud.
2. Review the information about your service in the email.
3. Click **Activate My Service**.
4. Fill out the form to sign up for your new Oracle Cloud Account.

You will be asked to:

- Create a new account name, which will be used to identify your Cloud Account.
- Provide your email address if prompted. You must provide the same email address at which you received your welcome email. Instructions for logging in to your new Cloud Account will be sent to this address. You'll be prompted for the email ID only if you don't already have a Cloud Account.
- Select a Default Data Region. If you need more information, click the **Data Regions** link below the field.
- Provide Cloud Account Administrator details. The person you specify here will be both a Cloud Account Administrator and a Service Administrator and can create other users as required. This person will manage and monitor services in the specified Cloud Account.
- When you have entered all the required information, click **Create Account** to submit your request for an Oracle Cloud Account.

After successful activation, you'll receive another email with your login credentials. Use this information to sign into your account and change your password on initial login.

Sign In to Your Cloud Account For the First Time

After you activate your account, you'll get a welcome email. The email provides you with your cloud account details and sign-in credentials. When all the services are provisioned in your account, you'll be notified on the My Services dashboard.

1. Open the welcome email and scroll down to the **Access Details** section.
2. Note the user name and the temporary password, and then click **Get Started with Oracle Cloud**.
3. Enter the user name and temporary password from the welcome email and click **Sign In**.
4. You'll be prompted to change your password the first time you sign in.

After you change the default password, you are directed to the Guided Journey. You can then either:

- Use the Guided Journey to learn about the services, tutorials, and other documentation available to help you get started with Oracle Cloud. See [Get Started with the Guided Journey](#).

OR


- Click **Dashboard** to go to the My Services Dashboard, where you can create a new Cloud service instance or explore the features of your Cloud Account.

Extend Your Cloud Account to Your Cloud at Customer Region

After you sign in to your Cloud Account with Universal Credits in an Oracle data region, you can then extend your subscription to include your Oracle Cloud at Customer data region.

1. Sign in to My Services.
2. Click the **Account Management** tile in the My Services dashboard and then select the **Account Management** tab to view your subscription details.

A list of services or entitlements in your account is displayed.

3. Locate the IaaS/PaaS service category.
4. From the  **Action** menu, click **Manage Data Regions**.
5. A new region shows up that represents the Oracle Cloud at Customer environment in your data center. The region has the format:

```
EXT_EXTSITE_001
```

For example:

```
EXTSITE_201806140022011001
```

Select this Cloud at Customer data region to extend your account to the new OCC rack in your data center.

6. Click **OK** to proceed.

You'll receive another welcome email with sign-in credentials to the selected data region after the services are provisioned in the new region and the process is complete. After you receive the credentials, you can log in to the new account and create users and service instances as required.

When you extend your subscription to another data region, Oracle Cloud automatically appends the selected data region name to your existing cloud account. This is your cloud account identity for the new data region. For example, if your primary data region is EMEA, your cloud account name is ABCComp1, and you extend your services to APAC (Asia Pacific) data region, then your new cloud account name for that region will be ABCComp1-APAC. You can also switch between your cloud accounts. See [Switching Between Accounts](#) for details.

When you sign in or switch to the identity domain for that region, you can access the services available there. Expand the IaaS/PaaS service category in the Account Management page to see what services are available in the new region.

Sign In to Your Cloud at Customer Data Region

After you extend your Oracle Cloud Universal Credits account to the Oracle Cloud at Customer data region, two additional emails are generated.

1. An activation email:

Use this email to activate the Oracle Cloud at Customer account. Click **Activate** and fill in a short form to activate the account.

2. A welcome email:

Use this email to sign in to your Oracle Cloud at Customer account. Click **Get Started with Oracle Cloud**. Enter the user name and the temporary password from the welcome email, and click **Sign In**.

After you sign in and change the default password, you are directed to the My Services Dashboard. A set of Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) are provisioned on Oracle Cloud at Customer by default. You can later request for additional IaaS and PaaS services that are available as part of your Universal Credits subscription.

5

Get Started with Your Cloud at Customer Account

As an Oracle Account Administrator, there are a few tasks you typically perform when you first get access to your Oracle Cloud at Customer subscription.

Topics

- [Explore the My Services Dashboard](#)
- [View Oracle Cloud Service Details](#)
- [Access a Cloud Service Console](#)
- [Create Instances](#)
- [About the Documentation for Oracle Cloud Services](#)

Explore the My Services Dashboard

Use the My Services Dashboard page to check the overall status of your purchased services and to perform basic administration tasks.


Topics

- [Basic Features of the My Services Dashboard](#)
- [The Welcome Section](#)
- [The Cloud Services Section](#)

Basic Features of the My Services Dashboard

The My Services Dashboard gives you access to basic information about your account, as well as system notifications and access to the various account management features.

Across the top of the screen, you'll see the account name above the **Dashboard** button. The **Dashboard** button always returns you to the dashboard.

A bell icon  displays important notifications, if any, at the top of the page. This is known as the Message Center and indicates important messages for the selected account.

Message summaries, if any, are automatically displayed in a pop-up window. Click **More Info..** for details on a specific message. Click **Show All Important Messages** to display previously hidden messages, if any. Click **Go to Notifications** to open the Notifications list page.

The user name that you used to sign in to My Services appears on the top right corner of the page. Click your user name to display a menu of additional features.

Click the **Users** button to create and manage additional Oracle Cloud Account users.

If you have access to more than one Oracle Cloud Account, use the **Cloud Account** menu to switch between accounts.

The Welcome Section

The Welcome section enables you to get started with activities that you would typically perform upon logging in to My Services.

You can expand or collapse the Welcome section by using the relevant buttons.

The following table describes the tiles in the Welcome section.

Tile	Description
Create Instance	<p>Click this tile to create service instances.</p> <p>Note: This tile is displayed only if the logged-in account or domain contains metered services or entitlements.</p> <p>The Create Instance dialog box displays 2 tabs, namely, Quick Start Services and All services. The Quick Start Services tab displays most popular or most used services such as Compute, Storage, Java, Database, Application Developer, Business Integration, Integration, or Mobile for which you can create instances.</p> <p>Note: This tab is active only if your logged-in domain contains these services and if you are the administrator for these services. If not, this tab is disabled.</p> <p>The All services tab displays all the other services. Click Create to create an instance for the selected service. You can also search for a service in the Create Instance dialog box.</p> <p>See Creating a Service Instance.</p>
Account Management	<p>Click this tile to view and manage your subscriptions and account usage. You can also assign an account and activate your services. See Managing Your Account and Subscriptions.</p>
Customize Dashboard	<p>Click this tile to customize your dashboard display. From the Customize Dashboard dialog box, you can control the tiles to be displayed on the dashboard by selecting one of the following:</p> <ul style="list-style-type: none"> • Automatic: This button is displayed only if enabled for the selected service type. If you do not select Show or Hide specifically, the system automatically displays or hides a tile based on the service instance count. Service tiles are displayed if instance count is greater than zero and if there are less than 16 tiles on the dashboard. • Show: Select this button to display the service tile on the dashboard. Purged and terminated services will not be displayed in the list. Services which are not yet active or still in progress won't be displayed on the dashboard. Only a maximum of 16 services can be displayed on the dashboard. • Hide: Click this button to hide the tile from the dashboard. Or, hover over the tile and click X. <p>Services within the Customize Dashboard dialog box are grouped into Platform Services, Applications, and Metered Services Categories and Service Entitlements. A service entitlement is displayed in 2 rows, the first row indicates the name of the entitlement; the second row displays the number of instances and a link to the entitlement detail page.</p>

The Cloud Services Section

Use the **Cloud Services** section in the dashboard to view the services available in your Cloud Account.





Note:

If you don't see a specific service you are looking for in the Cloud Services section of the page, click [Customize Dashboard](#) to be sure the service tile is not hidden.

The following table describes the key elements shown on the My Services Dashboard page.

Element	Description
Account summary section	<p>Account summary is displayed, according to the user's locale, below Welcome section.</p> <p>Account Summary section displays one or more of the following:</p> <ul style="list-style-type: none"> • For metered services, Cloud credit balances per service category, which are remaining credits that haven't expired (if available). Click the value to open the Estimated Account Balance page. • Charges to date for pay as you go metered services (if available). Click the value to open the Estimated Account Balance page. • Overage charges to date, if any, for service entitlements. Click the value to open the Estimated Account Balance page. • Resource usage for service entitlements. Only purchased resources with less than 50% of the remaining balance are displayed in this section. Click the value to open the Resource Quotas page. • For credit promotions, a flag is displayed indicating that the service is under promotion. Hover over the promotion flag to view promotion details. Remaining balance for promotion is also displayed in this section. Click on the amount to view the Estimated Account Balance page. <p>If the promotion has expired, then you can convert your promotion to a paid (Pay As You Go) account by clicking the Update Plan button. Hover over the button to see the Convert to Pay As You Go option that you selected when you signed up. Clicking the Update Plan button takes you to the Oracle Store, where you can make changes to your promotion.</p> <ul style="list-style-type: none"> • If you opted for monthly commit subscription, then the subscription category is displayed in this section along with the start and end dates of the committed month. <p>If the amount displayed for any of the services exceeds the space available, it is displayed as an ellipsis. Hover over it to see the full value.</p> <p>Note that the expired purchase amounts are excluded from the displayed values.</p>

Element	Description
Service tiles	<p>Each tile displays the service name and instance counts for up to three instance status categories.</p> <p>For non-metered services, a short display name of the service type is displayed. Icons on the tile indicate if you are an account administrator, service administrator, or both, for the selected service.</p> <p>You can change the order of the tiles displayed on the dashboard. Hover over the tile, then click and drag the dots at the top of the tile to drag the tile to the desired location on the dashboard.</p> <p>Use the + or – icons to expand or collapse the tiles.</p> <p>A status icon indicates the state of the service. For example, a tick mark (in green) indicates that the service is in active state.</p> <p>Instance status is indicated by color:</p> <ul style="list-style-type: none">• Active, ready, running, or soft-terminated instances (Green)• Instances that are new or not active. For example, Initialized, disabled, failed, or termination-in-progress (Yellow)• Canceled, terminated, suspended, corrupted, or unreachable instances (Red). <p>If the count is zero for any instance status category, then its category is not displayed on the tile.</p> <p>Hover over specific instance status category count to see the breakdown of status values.</p> <p>For nonmetered services, an instance count of 1 is always displayed, and in appropriate color, based on the service status.</p> <p>Services that are reclaimed or pending reclaim are also displayed on the dashboard but you can't view their corresponding service detail pages.</p>
	<p>Click the Action Menu icon on the service tile to select options for:</p> <ul style="list-style-type: none">• View Oracle Cloud Service Details• Access a Cloud Service Console Select this link to open a service instance console for applications or an administration console for all other service types. This action is disabled if the selected service is locked.• Viewing Account Balance Details in My Services in <i>Managing and Monitoring Oracle Cloud</i>. This page displays overage costs. This button is disabled for trial services. <p>Only actions that you can perform as a service administrator are available from the menu.</p>

Element	Description
	<p>To view metrics within the service tile, expand the tile. Click the name of the metric graph to view historical usage details. You can't view the details of purged or deleted services or entitlements.</p> <p>Click this icon to open the Select Content dialog box and select the related metrics (billing, business, or monitoring metrics, if available) that you want to display within the service tile. Instances that do not have any recent usage data are not displayed when you select monitoring metrics.</p> <p>Metric data is displayed according to the time zone set in the Preferences dialog box. If a time zone isn't specified, metrics are displayed according to the browser's time zone.</p> <p>You can select up to 4 metrics for entitlements and metered services. For other single-instance subscriptions, you can select up to 2 metrics for display.</p> <p>If a service has fewer than 2 metrics, then this icon is disabled. For test instances, a test flag is displayed.</p>

View Oracle Cloud Service Details

This section describes the elements on the service details page.

In My Services, the details page for a service:

- Displays status, uptime, and utilization data
- Lets you complete administration tasks, such as locking a service or associating services
- Provides links to the service console, the service instance, and the Oracle store

To view the details for a service from My Services:

1. Sign in to My Services.
2. Navigate to the appropriate service listing.
3. Click the service name to open the details page for that service.

At the top of the page, clicking the triangle to the left of **Service Details:** **<servicename>** displays general information about the service.

4. Click each tile on the left to view more information about the selected service. By default, the Overview tile is in focus.

 **Note:**

The number of tiles that are displayed, and the information that is provided on each tile varies from one service type to the next.

The following table provides a brief description of the more typical tiles that appear when you display the Service Details page.

Tile	Description
Overview	<p>Displays additional information about the service, including plan, service dates, subscription ID, and SFTP accounts.</p> <p>For some services, you can view a service status calendar, which shows the historical status (Month View), availability, or uptime of this service, either quarterly or yearly (Quarterly View or Year View). See <i>Monitoring Current and Historical Utilization for a Service</i> in <i>Managing and Managing Oracle Cloud</i>.</p>
Business Metrics/ Billing Metrics	<p>Displays the usage data collected for this service. The data collected depends on the type of service.</p> <p>You can also set alerts so you know immediately when billing metrics are nearing a specific threshold.</p> <p>See <i>Viewing Service Details in My Account for Metered Oracle Cloud Services</i> in <i>Managing and Managing Oracle Cloud</i>.</p>
Monitoring Metrics	<p>For some services such as Oracle Cloud Infrastructure Compute Classic, Oracle Java as a Service (Oracle JaaS), or Oracle Database as a Service (Oracle DBaaS), you can monitor real-time service usage data to help you determine whether the resource allocations for a service are underutilized or overutilized.</p> <p>You can also set alerts, and monitor current and historical usage data for service instances. The graphs are rendered in the time zone you set in the Preferences page.</p> <p>See <i>Monitoring Service Usage</i>.</p> <p>See <i>Monitoring Real-Time Usage Across Services</i>.</p>
Resource Quotas	<p>This tile is visible only when resources have been purchased. The right pane displays the type of resources, purchased limit or quota and the available balance of these resources.</p>

Access a Cloud Service Console

From the My Services Dashboard, you can access the individual services to which you've subscribed. For most Cloud Services, that means accessing the management console for the service.

To access a service console:

1. Log in the My Services Dashboard.
2. Click **Customize Dashboard** to make sure that the service you want to use is not hidden.
3. Locate the service tile in the Cloud Services section of the dashboard, and click the name of the service in the tile.

This displays the Service Details page.

4. On the Service Details page, locate the **Open Service Console** button.

The actual button name will vary, depending on the service. Also, some services do not have a management console or provide you with access to other features of the service from the Details page.

Create Instances

After you log in to the Oracle Cloud My Services Dashboard, you create instances of the Cloud services available in your Cloud Account.

To create a new instance:

1. Click **Create Instance** in the Welcome section of the My Services Dashboard.

2. In the resulting dialog box, select one of the Cloud Services.

Follow the instructions on the screen to get started. Depending on the service, you will be prompted to create a new instance or to learn more about the types of instances you can create.

For example, if you select **Compute Classic**, then you're directed to the first screen of a wizard, which guides you through the process of creating your first Compute Instance (or VM).

If you select **Database**, then you're directed to the Services page of Database Cloud Service console. From there, click **Create Service** to create your first Database instance.

About the Documentation for Oracle Cloud Services

The Oracle Cloud services available on Oracle Cloud at Customer are the same services available on Oracle Public Cloud. As a result, you can use the same Oracle Public Cloud documentation to learn about each Cloud service.

As you are using the documentation, you might see notes that indicate that specific features are not supported or behave slightly differently on Oracle Cloud at Customer. This is because there are sometimes restrictions or differences in the behavior of a service when they are used within a local customer data center, as opposed to a larger, central data center owned by Oracle.

6

Monitor Your Usage and Universal Credits Balance for Oracle Cloud at Customer

At any time, you can check your account balance and your current Oracle Cloud at Customer service usage. You can view your usage by region, by service, or by a specific time period.



Note:

To ensure that you see the most up to date information, Oracle recommends checking your Universal Credit in your Cloud Account using Oracle Public Cloud (OPC) url address links.

To check your account balance and usage, Oracle recommends that you sign in to your Oracle Cloud Account in an Oracle data region. From there, you can view your overall account usage and Universal Credits balance. Refer to the following topics for more information.

Topics

- [Sign In to Your Oracle Cloud Account to Check Your Balance](#)
- [Check Your Account Balance and Usage Summary](#)
- [Set an Alert to Monitor Your Account Balance](#)
- [Use the REST API to Check Your Account Balance](#)

Sign In to Your Oracle Cloud Account to Check Your Balance

When you sign up for a Universal Credits subscription on Oracle Cloud, you have access to services on your Oracle Cloud at Customer installation and to services available on Oracle Cloud. When checking your Cloud Account balance, Oracle recommends you sign in to your Cloud Account on Oracle Cloud to check your balance usage information.



Note:

To ensure that you see the most up to date information, Oracle recommends checking your Universal Credit in your Cloud Account using Oracle Public Cloud (OPC) url address links.

1. Locate the My Services URL and sign-in credentials for your Cloud Account on Oracle Cloud.
You can find this information in your initial Welcome email.
2. Enter the My Services URL in your browser and sign in to the My Services Dashboard.

From this Dashboard, you can create and manage services in your Oracle Cloud data region. To see the available services, click **Customer Dashboard** or **Create Instance**.

From this Dashboard, you can also check your account usage and perform account manage tasks, by clicking **Account Management**.

Check Your Account Balance and Usage Summary

You can view the up-to-date, estimated account balance details of your services or subscriptions in My Services, Account Management page.

To view your account balance details, click the **Account Management** tile on the My Services dashboard.

The Usage page displays the aggregated usage charges for individual services along with resource utilization and overages, if any. Select the category you want to view from the account type (Promotion, Monthly Flex) drop-down list. You'll see the following sections:

- **Usage Period:** Select a date range from the calendar to view usage details for that period. Date and time are displayed based on your time zone preferences. Usage charges and the currency are displayed for the selected date range. Overages, credit balance, expiry are also displayed if applicable.

For monthly flex, this section displays the monthly recurring prepaid subscription details along with the monthly usage and overage charges (if any). The active monthly billing period is displayed by default.

Use the **Scope** filter to view billing details pertaining either to your Cloud Account, your primary data region, or extended data regions.

- Select **Cloud Account** to view aggregated billing details of services across all data regions within the Cloud Account.
- Select the *primary data region* to view billing details of all services in that region.
- Select an *extended data region* to view the billing details of that region. However, you can view details only if you've logged in to the primary data region.

- **Usage Summary**, which provides details of all the resources in the service category, their usage quantity, charges and overages if any. Expand the service category to see the details. For cloud promotions, this section displays credits used and the remaining balance for the services in your promotion.

Note that you can filter the usage summary by using tags. These tags are shown in key=value pairs. For example, OCIService=Database. Click the **Filter by tags** text box to select the required tags. You can select multiple tags if required. Usage summary is shown based on all the tags you select. .

Download Your Account Balance and Usage Summary

After you view your account usage summary, you can save the usage details in the Account Management, Usage page.

To download the usage summary:

1. Sign in to My Services.
2. Click the **Account Management** tile.
3. In the Usage page, select the date range to view your usage summary and use tags if required, to show usage of specific services or resources in your cloud account.
4. Click the **Download as CSV** button.

The usage details are downloaded to a CSV file. You can then use this file to determine your account usage and take necessary action, if required.

Obtain Usage Data for Your Cloud at Customer Region

When you sign in to your Oracle Cloud Account in your default Oracle data region, you can view the usage data for the other data regions in your account. For example, if you are a Cloud at Customer user, you can view and download the current usage data for the services and resources you are using on your Cloud at Customer subscription.

To view the usage data for your Cloud at Customer data region:

1. Sign in to your Oracle Cloud Account in your default Oracle data region.
2. Click the **Account Management** tile on the My Services dashboard.
3. On the Account Usage page, set the **Usage Period**, if required.
4. Use the **Scope** filter to select the Cloud at Customer data region. This displays the Usage Summary for your Cloud at Customer data region.

You can also download the Usage Summary by clicking **Download as CSV**.

Set an Alert to Monitor Your Account Balance

You can monitor your account balance in your cloud account by generating alerts. This helps you determine whether to increase your account balance or continue with your purchased amount. The account balance is monitored vis-a-vis a defined usage limit (amount value).

You can configure rules to generate alerts when the account balance of metered services category reaches or exceeds the specified usage limit. If a soft limit is set, users are allowed to exceed the amount up to a system-defined value. If a hard limit is set, then a quota breach occurs when the specified limit is reached. When a quota breach occurs, you must increase your account balance to continue to use the services.

To create alert rules, do the following:

1. Sign in to My Services.
2. Select a service and click the service name to open the details page for the service.

The Overview page is in focus.


3. Click the **Billing Alerts** tab. Alert rules, if any, are listed in the Alert Rules section. The **Billing Alerts** tab is displayed based on the account type and user role. You can set alerts only for services that support it and only if you're a Cloud Account Administrator or a Service Administrator.
4. In the Alert Rules section, click **Create**.
5. In the Create Alert Rule dialog box, specify the following:
 - a. **Channel:** Email. This is selected by default and is read-only.

- b. **Service Category Name:** Displays the selected metered services category that this service belongs to and is read-only.
- c. **Limit Type:** Specify one of the following:
 - **Soft Limit:** You'll get an alert when the usage limit is reached, but you can still continue to use the resources.
 - **Hard Limit:** You'll get an alert when the usage limit is reached and a quota breach occurs. You can't create new instances as the service becomes suspended, however, you can still continue to use existing services or instances. You can specify only one hard limit alert rule for the purchased amount. If there's an existing system-created hard limit alert rule for the purchased amount, then you can't create another hard limit rule for the same amount. Also, you can't delete system-created hard limits.
- d. **Value Type:** Select either **Absolute** or **Percentage**. You can't specify **Percentage** for pay-as-you-go subscriptions.
- e. **Usage Limit:** Specify the usage limit. Limit value can't be more than what is allowed by the system depending on whether it's an absolute value or a percentage. You'll get an alert when the resource usage reaches this value. The limit must be greater than or equal to 1. For Universal Credit subscriptions, the overage value is the cumulated amount of all services in the order from the date of purchase, and is displayed below this field.
- f. Click **Done**.

Example 6-1 Configuring an Alert Rule to Monitor Account Balance

- Your purchased amount =100 USD
- Limit Type = Soft Limit
- Usage Limit = 50 USD, limit value cannot exceed more than 2 times the purchased amount per system configuration

Then, an alert is generated when you have used 50 USD, but you are still allowed to create instances up to 200 USD (2 times the purchased amount), because you specified a soft limit. If you specify a hard limit of 200 USD, then a quota breach occurs and you aren't allowed to create any more instances until you increase your account balance.

The alert rules are listed in the Alert Rules section in the Monitoring page. You can modify or delete alert rules here. To modify a rule, click **Modify** from the  **Action** menu and make the necessary changes to the limits. When modifying an alert rule, you can't change the limit types (either hard or soft) or the resource.

To delete a rule, click **Delete** from the  **Action** menu. You can't delete system-created alert rules.



Note:

You can modify or delete an alert rule only for resources that you have purchased.

Use the REST API to Check Your Account Balance

Another method of tracking metrics is to use the Oracle Cloud Metering API to monitor your cloud account and set alert thresholds to monitor the account balance.

To use the API, you must have the following:

- A paid or a promotional subscription to an Oracle Cloud service
- Sign-in credentials to access the Oracle Cloud Metering API
- Required roles to access the service and perform GET requests
- cURL application

For information on how to use the Oracle Cloud Metering API, see [Oracle Cloud Account Metering REST API](#) documentation.

7

Monitor Your Cloud Service Performance

You can monitor Oracle Cloud services and set alert thresholds for specific metrics in My Services.

Topics

- [View Performance Metrics for an Oracle Cloud Service](#)
- [Set an Alert for a Performance Metric](#)

View Performance Metrics for an Oracle Cloud Service

For some Oracle Infrastructure and Platform Cloud Services (Oracle IaaS/PaaS), you can monitor real-time service usage data to help you determine whether the resource allocations for a service are underutilized or overutilized. You can also set alerts, and monitor current and historical usage data for service instances. The graphs in this page are rendered in the time zone you set in the Preferences page.

To monitor usage metrics:

1. Sign in to My Services.
2. Navigate to the appropriate page and find the service for which you want more information.
3. Click the service name to open the details page for the service.

The Overview page is in focus.

4. Click the **Monitoring Metrics** tab to review usage data for the selected service. This page displays usage data in the form of graphs. The **Graph** tab displays the related graphs, if any. By default, one graph is displayed with the first metric from each available instance (maximum 3) plotted on the graph. However, each graph can display usage data for a maximum of 5 service instances.

In this page, you can monitor and customize the following:

- **Show Thresholds/Alerts:** Click the button to set alert rules to notify service administrators when a metric is either above or below defined thresholds. This button is displayed only when you select the **Graphs** tab. See [Set an Alert for a Performance Metric](#).
- **Viewport/Timeline graph:** The graph displayed at the top is called the viewport graph and shows usage data of the selected metric for the period specified in the timeline graph. Granular usage data is shown when you set the time range to an hour.

The timeline graph is displayed below the viewport graph and allows you to select a time frame by using a viewport slider. Drag the slider to select a time period. You can also drag the edges of the slider to increase or decrease the selected time period. Usage data of the metric is displayed in the viewport graph according to your selection in the timeline graph. Usage data is auto-refreshed every minute, when the

slider is positioned at the extreme right in the timeline graph. To hide the timeline graph, click **Hide Overview**.

- **From/To Dates:** You can also select specific dates and time from the calendar for a more precise view of metric usage data. By default, the viewport graph displays usage data for the last 2 hours and the timeline graph displays data up to 1 year before the current time, if available. The system calculates and displays usage data based on Coordinated Universal Time (UTC).
- **Add Graph:** You can select additional graphs to be displayed in this section by clicking the **Add Graph** button. This is useful when you want to plot and compare data with the same metric units. By default, 3 metric graphs are displayed. You can select up to a maximum of 4 metric graphs. To remove a metric from the display, click the X icon.
- **Add Metric:** You can select additional instances to be displayed within each graph by clicking the **Add Metric** button below the graphs. You can select up to a maximum of 5 instances. To remove a metric, click the X icon. Customization of the graph and instance display is saved and the same will be displayed when you view the Monitoring Metrics page the next time. However, any customizations to the viewport graph are not saved. Deleted instances are also displayed in the list and denoted by an asterisk. You can select a deleted instance to view its historical usage data, however, you can't select the same within the service tile on the My Services dashboard.

Set an Alert for a Performance Metric

For some services such as Oracle Cloud Infrastructure Compute Classic, Oracle Java Cloud Service, or Oracle Database Cloud Service, service administrators can configure rules to generate alerts when metrics exceed or are under specified thresholds for a specific time period. Service administrators will receive the alert notifications.

A single metric is monitored over a specific time period vis-a-vis a defined threshold value. An alert is sent to the service administrators when there is a change in the metric behavior for a sustained period of time that is defined by you. The threshold state can be either `true` (satisfied) or `false` (unsatisfied) depending on the change in the metric behavior.

An alert is sent only when the threshold state changes as follows:

- Metric is not within the defined threshold for a sustained period of time *and* the threshold state changes from either `true` to `false` or vice versa.
- Metric is within the defined threshold for a sustained period of time *and* the threshold state changes from either `true` to `false` or vice versa.

To create alert rules, do the following:

1. Sign in to My Services.
2. Navigate to the appropriate service listing.
3. Click the service name to open the details page for the service.
The Overview page is in focus.
4. Click the **Monitoring** tile to configure alerts for the selected service.
5. In the Alert Rules page, click **Add**.
6. In the Create Alert Rule dialog box, specify the following:

- a. **Channel:** Email. This is selected by default and is read-only.
- b. **Instances:** Select an instance from the list to monitor.
- c. **Criteria:** Select the metric to monitor. Select the average, the operators (>, <, <=, >=) and then enter an amount in the adjacent box. This is the threshold value. The amount must not be less than zero.
- d. **Duration:** Select the time period for which you want to monitor the metric.
- e. **Period Count to Satisfy:** Enter the number of consecutive times that the metric must meet the criteria you specified above to trigger an alert. The count must be greater than or equal to 1.
- f. **Period Count to Unsatisfy:** Enter the number of consecutive times that the metric must be within the threshold limit to trigger an alert. The count must be greater than or equal to 1.
- g. Click **Done**.

Example 7-1 Configuring an Alert Rule to Monitor CPU Usage

- Criteria = CPU Percentage (%) Average >=80
- Duration = 10 minutes
- Period Count to Satisfy = 3
- Period Count to Unsatisfy = 2

Then,

An alert is sent once when the metric:

- Crosses the specified threshold during 3 consecutive 10-minute periods and the threshold state changes from false to true.
- Remains below or is within the threshold limits for 2 consecutive 10-minute periods (after a corrective action is taken) and the threshold state changes again from true to false.

Another alert is *not* sent:

- If a metric remains above the threshold for more than 3 consecutive 10-minute periods and there is no change in the threshold state.
- If a metric remains below or is within the threshold for more than 2 consecutive 10-minute periods and there is no change in the threshold state.

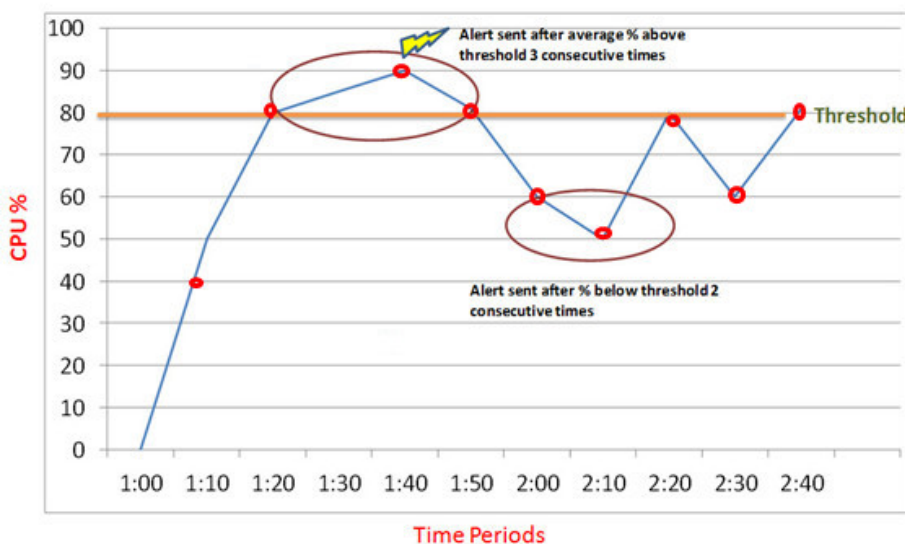
The table below summarizes when an alert is triggered based on the specified criteria.

Time Period	Metric (CPU Usage % >=80)	Threshold Value (Above/Below/Equal)	Threshold State (True/False)	Alert Sent (Y/N)
1:00	Threshold Created	0	False	N
1:10	50	Below	False	N
1:20	80	Equal	True	N
1:30	85	Above	True	N
1:40	90	Above	True	Y (Avg>=80 for 3 consecutive 10-minute periods)

Time Period	Metric (CPU Usage % >=80)	Threshold Value (Above/Below/Equal)	Threshold State (True/False)	Alert Sent (Y/N)
1:50	81	Above	True	N
2:00	60	Below	False	N
2:10	50	Below	False	Y (Avg<80 for 2 consecutive 10-minute periods)
2:20	80	Equal	True	N
2:30	60	Below	False	N
2:40	81	Above	True	N

Here is an illustration of the above example:

Alert Rule Illustration



The alert rules are listed in the Alert Rules page. To delete a rule, click **Remove** from the **Action** menu.

8

Create and Manage Users for Oracle Cloud at Customer

One of the important tasks you perform as an Oracle Cloud Account Administrator on Oracle Cloud at Customer is creating and managing additional users. You can then assign specific roles and privileges to each user.

Topics

- [Sign In to Your Oracle Cloud at Customer Account to Create Users](#)
- [About the Users Page in a Cloud Account](#)
- [Create a New Cloud Account User](#)
- [Create Groups](#)
- [Assign Cloud Account Roles to a User](#)
- [Import a Batch of Users into a Cloud Account](#)

Sign In to Your Oracle Cloud at Customer Account to Create Users

When you create users to manage your Oracle Cloud at Customer services, be sure you are signed in to your Oracle Cloud at Customer data region.

1. Locate the My Services URL and sign-in credentials for your Cloud Account on Oracle Cloud at Customer.

You can find this information in your Welcome email you received after extending your Oracle Cloud subscription to Oracle Cloud at Customer .

2. Enter the My Services URL in your browser and sign in to the My Services Dashboard.

From this Dashboard, you can create and manage services in your Oracle Cloud at Customer data region. To see the available services, click **Customize Dashboard** or **Create Instance**.

From this Dashboard, you can also create and manage Cloud at Customer users.

About the Users Page in a Cloud Account

If you're using a Cloud Account with Identity Cloud Service, then you can perform the following tasks from the My Services **Users** tab.

- Create a new Cloud Account user
- Assign Cloud Account roles to a user
- Change your password
- Modify or remove user accounts

- Go to **Identity Console** to access the complete set of Oracle Identity Cloud Service features for advanced user management.

For details on modifying or removing user accounts, see Managing Users with IDCS Cloud Accounts in *Managing and Monitoring Oracle Cloud*.

For information about using the complete set of Oracle Identity Cloud Service features, see Managing Oracle Identity Cloud Service Users in *Administering Oracle Identity Cloud Service*.

Create a New Cloud Account User

If you are a Cloud Account administrator, you can create user accounts from the **Users** tab in My Services.

To create a user account:

1. Sign in to My Services with your sign-in credentials.
2. In the dashboard, click **Users**, and then click **Add**.
3. In the Add User Details dialog box, enter the first name, last name, and email address of the user.

To use the email address as the user name, leave the check box selected. If you'd like to use a user name other than the email address, clear the check box and provide a user name.

4. Click **Finish**.

Later, after you add multiple users, you can organize those users into groups, so you can quickly assign roles and privileges to the group.

The user account is created and the Details page is displayed.

For more information, see Managing Oracle Identity Cloud Service Users in *Administering Oracle Identity Cloud Service*.

Create Groups

You can create groups in Oracle Cloud at Customer.

1. Log in to your Cloud Account and go to **My Services**.
2. Click **Users**.
3. Go to the left navigation pane.
4. Click **Groups**.
5. Click **Add**.
6. In the **Name** and **Description** fields of the **Add Group** window, enter the name and descriptive information about the group.

WARNING:

Ensure that you specify the group name without space. Group names can contain alphanumeric characters, underscores, and dashes only.

7. To allow users to request access to this group, click **User can request access**.
8. To assign user accounts to the group, go to step 6. Otherwise, click **Finish**.
9. Click **Next**.
10. Select the check box for each user account that you want to assign to the group, and then click **Finish**.

 **Tip:**

To search for user accounts to assign to the group, in the search field, enter all or part of the beginning of the user names, first names, or last names of the user accounts that you want to locate, and then press **Enter**.

Assign Cloud Account Roles to a User

After you create additional user accounts, you can assign roles and delegate administrative responsibilities for these accounts.

Users can be assigned both administrative roles and application-specific roles. For example, if you want a particular user to be an administrator for Oracle Cloud Infrastructure Compute Classic, then assign the Application Administrator role to the account, and then assign the Compute Classic **Compute Operations** application-specific role to the account.

For more information, see:

- [Assigning Administrative Roles](#)
- [Assigning Application-Specific Roles](#)
- [Selecting Application-Specific Roles for Each Service](#)

Assigning Administrative Roles

To assign cloud account administrative roles to a user, do the following:

1. Sign in to My Services with your sign-in credentials.
2. In the dashboard, click **Users**.
3. From the side navigation bar, click the **Security**, and then click **Administrators**.

A list of available roles is displayed. For example, Identity Domain Administrator, Security Administrator, or Application Administrator.


4. Expand the role that you wish to assign to the user.
5. To add a user account to an administrator role, click **Add**.
6. Select the user and then click **OK**.

The selected user is granted the corresponding administrative privileges. See *Adding or Removing a User Account from an Administrator Role in Administering Oracle Identity Cloud Service*.

Assigning Application-Specific Roles

To assign application-specific roles to a user, do the following:

1. Sign in to My Services with your sign-in credentials.

2. In the dashboard, click **Users**.
3. From the side navigation bar, click **Applications**.
A list of relevant applications is displayed. Each application corresponds to an Oracle Cloud service available in your Oracle Cloud Account.
4. Select an application from the list and then select the **Application Roles** tab.
For example, if you want to allow a user to administer the Compute Classic, select the application with the name that begins with `Compute`, and then select **Application Roles**.
5. For each of the applicable predefined roles, click  **Action** and then select **Assign Users**.
6. Select one or more users from the Role window and then click **Assign**.

Selecting Application-Specific Roles for Service Administrators

Each Oracle Cloud service available on Oracle Cloud at Customer has a set of application-specific roles. For example, if you want a user to create and manage Compute instances or orchestrations, assign that user to the Compute Classic **Compute_Operations** role.

For more information about the application-specific roles available for some of the key Oracle Cloud services, refer to the following table. If the service you are using is not included in the table, locate the documentation for the service on the [Oracle Help Center](#). Each service guide includes a standard topic that lists and describes the roles for that service.

Cloud Service	More Information
Oracle Cloud Infrastructure Compute Classic	About Compute Classic Roles
Oracle Java Cloud Service	About Oracle Java Cloud Service Roles and User Accounts
Oracle Database Cloud Service	About Database Cloud Service Roles and Users

Import a Batch of Users into a Cloud Account

If you are cloud account administrator, you can batch import user accounts using a comma-separated values (CSV) file.

Before importing user accounts, you must first create a CSV file that is properly formatted for the import process. The CSV file is a simple text file in a tabular format (rows and columns). The first row in the file, which defines the columns (fields) in your table, must have these exact column headings such as First Name, Last Name, Work Email, and User ID.

To import user accounts:

1. Create a CSV file using any standard spreadsheet application. For each user account, create a new row (line) and enter data into each column (field). Each row equals one record.
2. Save your file in a CSV format.
3. Sign in to My Services with your sign-in credentials.
4. In the dashboard, click **Users** and then click **Import**.

5. In the Import Users dialog box, click **Browse** to locate and select the CSV file that contains the user accounts to import. You can also download a sample CSV file for your reference and use.
6. Click **Import**.
 - If the import job can be processed immediately, a dialog box appears with the Job ID link. You can review the details by clicking the link.
 - If the job can't be processed immediately, a Schedule ID is provided. Use the Schedule ID to search for the job in the **Jobs** page. The job appears in this page after the import process is complete.

See Importing user Accounts in *Administering Oracle Identity Cloud Service*.

Part II

Getting Started with IaaS and PaaS Services

This part provides step-by-step procedures to guide you through selected basic and advanced use cases for the IaaS and PaaS cloud services that are available on Oracle Cloud at Customer. The purpose is to help new users learn how to use these IaaS and PaaS cloud services in the initial days after the machine is activated.



Note:

Some of the procedures described in this part might not cover every feature that you see in the service interfaces. This part supplements the existing service-specific documentation, which you should continue to use as the exhaustive reference for the services available on Oracle Cloud at Customer.

Topics

- [Compute Classic: Basic Tasks](#)
- [Compute Classic: Advanced Tasks](#)
- [Compute Classic: Using the REST API](#)
- [Object Storage Classic: Managing Containers and Objects](#)

9

Compute Classic: Basic Tasks

Topics

- [Create an Oracle Cloud User with the Required Roles](#)
- [Generate an SSH Key Pair](#)
- [Create an Oracle Linux Instance](#)
- [Create an Oracle Linux Instance Using a Nonpersistent Boot Disk](#)
- [View Details of an Instance](#)
- [Enable SSH Access to a VM](#)
- [Log In to a VM Using SSH](#)
- [Add an SSH-Enabled User](#)
- [Reboot an Instance](#)
- [Shut Down and Restart an Instance](#)
- [Monitor Metrics for Your VMs](#)
- [Change the Shape of an Instance](#)
- [Create a Storage Volume](#)
- [Attach a Volume to a VM](#)
- [Mount a Volume](#)
- [Retrieve Predefined Instance Metadata](#)
- [Delete and Re-create an Instance](#)

Create an Oracle Cloud User with the Required Roles

If you're an account administrator or an identity domain administrator, then you can create user accounts.

1. Sign in to the Identity Cloud Service web console.
 - a. Sign in to **My Services**, as a user who has the *identity administrator* role.
 - b. On the **Dashboard** page, locate the **Identity Cloud** tile, and click **Identity Cloud**.
If the **Identity Cloud** tile is not displayed, then click **Customize Dashboard**, locate **Identity Cloud**, and click **Show**.
The Identity Cloud Service overview page is displayed.
 - c. Scroll down and click **Open Service Console**, near the lower-right corner of the page.
The Identity Cloud Service console is displayed.
2. Add a group.

- a. Expand the navigation pane on the left, and click **Groups**.
 - b. Click **Add Group**
 - c. In the **Name** and **Description** fields of the Add Group dialog box, enter a name and description for the group.
 - d. Click **Finish**.
3. Assign the required application roles to the group.
 - a. In the navigation pane, and click **Applications**.
 - b. Locate (or search for) the **Compute** application, and click it.
 - c. On the resulting page, click the **Application Roles** tab.
 - d. Locate the **Compute.Compute_Operations** role.
 - e. From the actions menu for the role, select **Assign Groups**.
 - f. Select the group that you created earlier.
 - g. Click **Assign**.
 - h. Repeat steps 3b through 3g for the following application roles:

Application	Role
Storage	Storage_Administrator
JAAS	JaaS_Administrator
DBAAS	DBaaS_Administrator

4. Add a user.
 - a. In the navigation pane, and click **Users**.
 - b. On the **Users** page, click **Add**.
 - c. In the **First Name** and **Last Name** fields, enter the user's first and last name.
 - d. In the **User Name / Email** field, enter the email address of the user.
 - e. Leave the **Use the email address as the user name** check box selected.
 - f. Click **Next**.
 - g. Select the group that you created earlier.
 - h. Click **Finish**.

Oracle sends a welcome email to the user. The user must follow the activation instructions in the email.

Generate an SSH Key Pair

To access your Compute instances using SSH, generate an SSH key pair, associate the public key with your instances, and use the private key to log in to the instances using SSH.

 **Caution:**

Keep your SSH keys secure. Lay down policies to ensure that the keys aren't lost or compromised when employees leave the organization or move to other departments. If you lose your private key, then you can't access your instances. For business continuity, ensure that the SSH keys of at least two IT system administrators are added to your instances.

Topics

- [Generate an SSH Key Pair on UNIX and UNIX-Like Systems](#)
- [Generate an SSH Key Pair on Windows](#)

Generate an SSH Key Pair on UNIX and UNIX-Like Systems

Use the following procedure to generate an SSH key pair on UNIX and UNIX-like systems:

1. Run the `ssh-keygen` command.

You can use the `-t` option to specify the type of key to create.

For example, to create an RSA key, run:

```
ssh-keygen -t rsa
```

You can use the `-b` option to specify the length (bit size) of the key, as shown in the following example:

```
ssh-keygen -b 2048 -t rsa
```

2. The command prompts you to enter the path to the file in which you want to save the key.

A default path and file name are suggested in parentheses. For example: `/home/user_name/.ssh/id_rsa`. To accept the default path and file name, press **Enter**. Otherwise, enter the required path and file name, and then press **Enter**.

3. The command prompts you to enter a passphrase.

The passphrase is not mandatory if you want to log in to an instance created using an Oracle-provided image. However, it is recommended that you specify a passphrase to protect your private key against unauthorized use.

 **Note:**

With some images provided on Oracle Marketplace, the use of a passphrase might be mandatory.

4. When prompted, enter the passphrase again to confirm it.

The command generates an SSH key pair consisting of a public key and a private key, and saves them in the specified path. The file name of the public key is created automatically by appending `.pub` to the name of the private key file. For example, if the file name of the SSH private key is `id_rsa`, the file name of the public key would be `id_rsa.pub`.

Make a note of the path and file names of the private and public keys. When you create an instance, you must specify the SSH public key value. When you log in to an instance, you

must provide the path to the corresponding SSH private key and you must enter the passphrase when prompted.

Generate an SSH Key Pair on Windows

You can generate an SSH key pair on a Microsoft Windows machine by using an application such as PuTTY. See the tutorial, [Creating SSH Keys for Use with Oracle Cloud Services](#).

Create an Oracle Linux Instance

Create an Oracle Linux instance using the Create Instance wizard.

1. Sign in to Oracle Cloud My Services.
2. Click the ☰ menu at the upper left corner of the page and select **Compute Classic**.
3. On the Instances page, click **Create Instance**.
The Create Instance wizard starts.
4. Click **Customize**.
5. On the Image page, select the image that you want to use, and click the right arrow button.
6. On the Shape page, select an appropriate shape for your instance, and click the right arrow button.
The shape determines the number of CPUs and RAM that your instance will have.
7. On the Instance page, select or enter the following. Leave the other fields at the default values.
 - a. **Name**: Enter an appropriate name, or retain the default name.
 - b. **Label**: Enter a label to help identify the instance, or retain the default.
 - c. **SSH Keys**:
 - i. Click **Add SSH Public Key**.
 - ii. Enter a name for the SSH public key.
 - iii. Click **Select File** and navigate to the folder where your SSH public key is saved, or paste the public key in the **Value** field.
 - iv. Click **Add**.Click the right arrow button.
8. On the Network page, select or enter the following:
 - a. **Network Options**: Select **IP Network**.
 - b. **IP Network Options**: Click **Configure Interface**.
 - c. In the Configure IP Network Interface dialog box, select or enter the following and then click Save:
 - **Interface**: Select the required interface, say **eth1**.
 - **IP Network**: Specify the IP network that you want to add this interface to.
 - i. Click **Create IP Network**.

- ii. Enter a name for the IP network.
 - iii. Enter the required IP address prefix for the IP network, in CIDR format. For example, enter 192.168.0.1/24.
 - iv. Enter a description and tags for the IP network, if required, and then click **Create**.
- d. **Public IP Address:** Select **Auto Generated**.
9. Click **Save**.
 10. Click the right arrow button to go to the next page.
 11. On the Storage page, accept the default settings and click the right arrow button.
 12. On the Review page, verify the information that you've entered and then click **Create**.
 13. Wait for your instance to be created. To monitor the status, go to the **Orchestrations** tab. Look for the orchestration that has the same name as your instance.


When the status of the orchestration is **Ready**, your instance is ready for use.

Create an Oracle Linux Instance Using a Nonpersistent Boot Disk

To create an instance that you can take a snapshot of and create clones from later, you must set up the instance to use a nonpersistent boot disk.

1. Sign in to Oracle Cloud My Services.
2. Click the ☰ menu at the upper left corner of the page and select **Compute Classic**.
3. On the Instances page, click **Create Instance**.
The Create Instance wizard starts.
4. Clicking **Customize**.
5. On the Image page, select the image that you want to use, and click the right arrow button.
6. On the Shape page, select an appropriate shape for your instance, and click the right arrow button.
The shape determines the number of CPUs and RAM that your instance will have.
7. On the Instance page, select or enter the following. Leave the other fields at the default values.
 - a. **Name:** Enter an appropriate name, or retain the default name.
 - b. **Label:** Enter a label to help identify the instance, or retain the default.
 - c. **SSH Keys:**
 - i. Click **Add SSH Public Key**.
 - ii. Enter a name for the SSH public key.
 - iii. Click **Select File** and navigate to the folder where your SSH public key is saved, or paste the public key in the **Value** field.
 - iv. Click **Add**.

Click the right arrow button.

8. On the Network page, select or enter the following:
 - a. **Network Options:** Select **IP Network**.
 - b. **IP Network Options:** Click **Configure Interface**.
 - c. In the Configure IP Network Interface dialog box, select or enter the following and then click **Save**:
 - **Interface:** Select the required interface, say **eth1**.
 - **IP Network:** Specify the IP network that you want to add this interface to.
 - i. Click **Create IP Network**.
 - ii. Enter a name for the IP network.
 - iii. Enter the required IP address prefix for the IP network, in CIDR format. For example, enter `192.168.0.1/24`.
 - iv. Enter a description and tags for the IP network, if required, and then click **Create**.
 - d. **Public IP Address:** Select **Auto Generated**.
9. Click **Save**.
10. Click the right arrow button to go to the next page.
11. On the Storage page, remove the default boot disk. From the  menu, select **Remove**. A nonpersistent boot disk is used to boot your instance..
12. On the Review page, verify the information that you've entered and then click **Create**.
13. Wait for your instance to be created. To monitor the status, go to the **Orchestrations** tab. Look for the orchestration that has the same name as your instance.

When the status of the orchestration is **Ready**, your instance is ready for use.


View Details of an Instance

After creating instances in Oracle Cloud Infrastructure Compute Classic, you can view a list of your instances and get details of each instance.

1. Sign in to the Compute Classic console.
2. The Instances page shows a list of instances, along with information about each instance.

Tip:

You can filter the list of instances according to their category or status. To list instances with a specific status (such as running, error, or stopped), click the **Show** menu and select the appropriate filter. To view instances of a specific category (such as PaaS, IaaS, or personal), click the **Category** menu and select the appropriate filter.

3. Go to the instance that you want to view. From the  menu, select **View**.

The instance details page shows all the details of the selected instance, such as the public and private IP addresses associated with it and details of interfaces added to IP networks. You can stop, start, or reboot the instance by clicking the appropriate icon at the top of the page. This page also displays the orchestration used to create the instance, and the storage volumes, security lists, and SSH keys associated with it. You can add or remove storage volumes and security list from this page.

Enable SSH Access to a VM

This procedure is for a VM that's attached to an IP network. SSH traffic to such VMs is controlled by security rules that you create.

1. Sign in to the Compute Classic console.
2. Identify the vNICset that contains the IP network interface of the VM for which you want to enable SSH access.
 - a. On the **Instances** tab, locate the VM for which you want to enable SSH access, and click its name.
 - b. In the **IP Network Interfaces** section, locate the vNIC for which you want to enable SSH access.
 - c. Note the name of the **vNICset** that you want to specify in the security rule.
If multiple vNICsets are associated with the vNIC, then pick the vNICset that's appropriate for your needs.
 - d. Return to the main page of the web console, by clicking the **Instances** link near the top of the page.
3. Identify the ACL that's applied to the vNICset.
 - a. Click the **Network** tab.
 - b. Expand **IP Network**, and click **Virtual NIC Sets**.
 - c. Locate the vNICset that you identified earlier, and note the name of the ACL applied to it.
If multiple ACLs are applied to the vNICset, then pick the ACL that should contain the security rule you're going to create.
4. Create a security protocol for SSH requests.
 - a. In the navigation pane, click **Security Protocols**.
 - b. Click **Create Security Protocol**.
 - c. In the **Create Security Protocol** dialog box, select or enter the following information:
 - **Name**: Enter `ssh`.
 - **IP Protocol**: Select **TCP**.
 - **Destination Port Set**: Enter **22**.
 - d. Leave the other fields at the default values.
 - e. Click **Create**.
5. Create a security rule.
 - a. In the navigation pane, click **Security Rules**.
 - b. Click **Create Security Rule**.

- c. In the **Create Security Rule** dialog box, select or enter the following information:
 - **Name:** Enter a name for the security rule.
 - **Status:** Select **Enabled**.
 - **Type:** Select **Ingress**.
 - **Access Control List:** Select the ACL that you identified.
 - **Security Protocols:** Select the protocol that you created.
 - **Destination vNICset:** Select the vNICset that you identified.
- d. Leave the other fields at the default values.
- e. Click **Create**.

You can now connect to the VM by using `ssh`. See [Log In to a VM Using SSH](#).

Log In to a VM Using SSH

Connect from UNIX and UNIX-Like Systems

You can log in to an Oracle-provided Oracle Linux instance as the default user, `opc`. The `opc` user has `sudo` privileges.

You can use SSH to log in to your instance as the default user, `opc`, by using the following command:

```
ssh opc@ip_address -i private_key
```

In this command, `ip_address` is the public IP address of the instance, and `private_key` is the full path and name of the file that contains the private key corresponding to the public key associated with the instance that you want to access.

If an error occurs, see [Can't connect to an instance using SSH in *Using Oracle Cloud Infrastructure Compute Classic*](#).

When you're logged in as the default user, `opc`, use the `sudo` command to run administrative tasks.

Connect from Windows

You can log in to an Oracle-provided Oracle Linux instance as the default user, `opc`. The `opc` user has `sudo` privileges. If you're using a Windows host, you can use PuTTY or any other similar client to connect to your instance using SSH.

1. Run the PuTTY program.

The PuTTY Configuration window is displayed, showing the Session panel.
2. In **Host Name (or IP address)** box, enter the public IP address of your instance.
3. Confirm that the **Connection type** option is set to **SSH**.
4. In the Category tree, expand **Connection** if necessary and then click **Data**.

The Data panel is displayed.
5. In **Auto-login username** box, enter `opc`.
6. Confirm that the **When username is not specified** option is set to **Prompt**.

7. In the Category tree, expand **SSH** and then click **Auth**.
The Auth panel is displayed.
8. Click the **Browse** button next to the **Private key file for authentication** box. Navigate to and open the private key file that matches the public key that is associated with your instance.
9. In the Category tree, click **Session**.
The Session panel is displayed.
10. In the **Saved Sessions** box, enter a name for this connection configuration and click **Save**.
11. Click **Open** to open the connection.
The PuTTY Configuration window is closed and the PuTTY window is displayed.
12. If this is the first time you are connecting to an instance, the PuTTY Security Alert window is displayed, prompting you to confirm the public key. Click **Yes** to continue connecting.

If an error occurs, see *Can't connect to an instance using SSH* in *Using Oracle Cloud Infrastructure Compute Classic*.

When you're logged in as the default user, `opc`, use the `sudo` command to run administrative tasks.

Add an SSH-Enabled User

If you've created your instance using an Oracle-provided Oracle Linux image, then you can use SSH to access your Oracle-provided Oracle Linux instance from a remote host as the `opc` user. After logging in, you can add users on your instance.

1. Generate an SSH key pair for the new user. See [Generate an SSH Key Pair](#).
2. Copy the public key value to a text file. You'll use this key later in this procedure.
3. Log in to your instance. See [Log In to a VM Using SSH](#).
4. Become the `root` user.

```
sudo su
```

5. Create the new user:

```
useradd new_user
```

6. Create a `.ssh` directory in the new user's home directory.

```
mkdir /home/new_user/.ssh
```

7. Copy the SSH public key that you noted earlier to the `/home/new_user/.ssh/authorized_keys` file.

```
echo "key" > /home/new_user/.ssh/authorized_keys
```

Here, `key` is the SSH public key value from the key pair that you generated earlier, enclosed in double quotation marks.

8. Add the new user to the list of allowed users in the `/etc/ssh/sshd_config` file on your instance, by editing the `AllowUsers` parameter, as shown in the following example:

```
AllowUsers opc myadmin
```

In this example, the `AllowUsers` parameter already had the `opc` user. The `myadmin` user has now been added.

9. Change the owner and group of the `/home/username/.ssh` directory to the new user:

```
chown -R new_user:group /home/new_user/.ssh
```

10. Restart the SSH daemon on your instance.

```
/sbin/service sshd restart
```

11. To enable `sudo` privileges for the new user, edit the `/etc/sudoers` file by running the `visudo` command.

In `/etc/sudoers`, look for the following line:

```
%opc ALL=(ALL) NOPASSWD: ALL
```

Add the following line right after the preceding line:

```
%group_of_new_user ALL=(ALL) NOPASSWD: ALL
```

You can now log in as the new user:

```
ssh new_user@ip_address -i private_key
```

In this command, `ip_address` is the public IP address of the instance, and `private_key` is the full path and name of the file that contains the private key corresponding to the public key that you added to the `authorized_keys` file earlier in this procedure.


If an error occurs, see [Can't connect to an instance using SSH in *Using Oracle Cloud Infrastructure Compute Classic*](#).

Use the `sudo` command to run administrative tasks.

Reboot an Instance

After your instance is running, if required, you can reboot your instance from the web console.

When you reboot an instance, data on storage volumes (whether persistent or nonpersistent) isn't lost. Your instance also retains all its configuration information, such as its public IP address and storage volumes that were attached and mounted on the instance.

1. Sign in to the Compute Classic console.
2. On the Instances page, go to the instance that you want to reboot. From the  menu, select **Reboot**.
The Reboot Instance dialog box appears.
3. (Optional.) If the instance hangs after it starts running, select the **Hard Reboot** check box to perform a hard reset of the instance.
4. Click **Yes** to reboot the instance.

Shut Down and Restart an Instance

If you created an instance using a persistent bootable storage volume, then, if you don't need the instance, you can shut down the instance. However, the instance isn't

deleted. After shutting down an instance, you can restart the instance later, without losing any of the instance data or configuration.

 **Note:**

To learn what happens when you shut down and restart an instance, see *Shutting Down and Restarting an Instance* in *Using Oracle Cloud Infrastructure Compute Classic*.

1. Sign in to the Compute Classic console.
2. On the Instances page, go to the instance that you want to stop. From the ☰ menu, select **Shut Down**.
While the instance is being shut down, its status changes to **Stopping**. When the instance has shut down, it continues to be listed on the Instances page with the status **Stopped**.
3. After the instance has shut down, to start the instance again, on the Instances page, go to the instance that you want to restart. From the ☰ menu, select **Start**.

Monitor Metrics for Your VMs

You can view real-time metrics for your Compute VMs.

1. Sign in to Oracle Cloud My Services.
2. Click the ☰ menu at the upper left corner of the page and select **Monitoring**.
The Monitoring Metrics page is displayed. By default, it shows one graph. To add a graph, click **Add Graph**.
In each graph, you can view data for the metrics and instances that you select and for a period that you define.
3. To adjust the period, use the **From** and **To** fields below the graph.
4. In the **Instance** field, select the VM for which you want to view metrics.

 **Tip:**

To view metrics for multiple VMs, create a group containing those VMs (in the **Groups** tab), and then select that group in the **Instance** field.

5. In the **Metric** field below the graph, select the metric that you want to view.

 **Note:**

To add another metric to the graph, click **Add Metric**.


- **CPU (%)**: Indicates CPU utilization in percentage.

- **Iostat Read (sectors):** Indicates the average number of sectors read, in I/O operations per second.
- **Iostat Write (sectors):** Indicates the average number of sectors written, in I/O operations per second.
- **Memory % (agent) %:** Indicates memory utilization in percentage, as reported by the OPC agent. The memory utilization metrics reported by this agent are more accurate than the memory utilization reported by the Memory Percent metric.
- **Memory (agent) (KB):** Indicates memory utilization in kilobytes, as reported by the OPC agent. The memory utilization metrics reported by this agent are more accurate than the memory utilization reported by the Memory Usage metric.
- **Memory Percent (%):** Indicates memory utilization, in percentage.
- **Memory Usage (KB):** Indicates memory utilization, in kilobytes.
- **Network Rcvd (B/s):** Indicates the average network traffic received by the VM, in bytes per second.
- **Network Sent (B/s):** Indicates the average network traffic sent by the VM, in bytes per second.



Change the Shape of an Instance


A **shape** is a resource profile that specifies the number of OCPUs and the amount of memory to be allocated to an instance in Oracle Cloud Infrastructure Compute Classic. The shape determines the type of disk drive that your instance uses.

You specify the shape of an instance while creating the instance. However, if your instance is managed by an orchestration, then you can change the shape of an instance even after the instance has been created. This is useful if you find that your application workload has increased and you would like to add OCPUs and memory to your instance.

1. Sign in to the Compute Classic console.
2. On the Instances page, identify and note the name of the instance that you want to update.
3. Go to the orchestration of the instance for which you want to change the shape.
4. From the  menu, select **Suspend**.
5. In the Suspend Orchestration confirmation window, click **Yes**.

The status of the orchestration changes to **Suspending**. After all the nonpersistent objects have been deleted, the status of the orchestration changes to **Suspended**.

6. After the orchestration status changes to **Suspended**, from the  menu, select **Update**.
7. On the orchestrations details page, in the Instance section, go to the instance that you want to modify. From the  menu, select **Properties**.
8. In the Object Properties dialog box, ensure that the **Persistent** check box isn't selected. If it is selected, deselect it, then click **Update**. This ensures that the status of the instance changes to **Inactive**.

9. On the orchestrations details page, in the Instance section, go to the instance that you want to modify. From the  menu, select **Update**.
10. In the **Shape** field, select the shape that you want to use for the VM.
11. Click **Update**.

The orchestration is updated with the specified shape.

12. Start the orchestration.

The orchestration is started and the instance is re-created using the specified shape.

To verify the shape your instance uses, you can view the appropriate orchestration. Alternatively, go to the Instances page and view the details of the instance.

Create a Storage Volume

A **storage volume** is a virtual disk that provides persistent block storage space for instances in Oracle Cloud Infrastructure Compute Classic. You can create storage volumes and attach them to instances to provide block storage capacity for storing data and applications. You can also associate a storage volume with a machine image, and then use the storage volume as the boot disk for an instance.

1. Sign in to the Compute Classic console.
2. Click the **Storage** tab.
3. Click **Create Storage Volume**.
4. Select or enter the required information:
 - Enter a name for the storage volume.
 - Enter the volume size, in GB.Leave the other fields at the default values.


5. Click **Create**.

While the new storage volume is being created, the **Status** field for the storage volume shows **Initializing**.

When the storage volume is ready, the **Status** field changes to **Online**.

Attach a Volume to a VM

You can provide or increase block storage capacity for an instance by attaching storage volumes.

1. Sign in to the Compute Classic console.
2. Click the **Storage** tab.
3. Identify the storage volume that you want to attach. From the  menu, select **Attach Instance**.
4. Select the instance to which you want to attach the volume.
5. The **Attach as Disk #** field is filled automatically with the next available index at which the volume can be attached. You can leave this field at the automatically selected disk number or enter a higher number up to 10.

The disk number that you specify here determines the device name. The disk attached at index 1 is named `/dev/xvdb`, the disk at index 2 is `/dev/xvdc`, the disk at index 3 is `/dev/xvdd`, and so on.

Make a note of the disk number. You'll need it later when you mount the storage volume on the instance.

6. Click **Attach**.

After attaching a storage volume to an instance, to access the block storage, you must mount the storage volume on your instance. See [Mount a Volume](#).

Mount a Volume

To access a storage volume, you must attach it to your instance and mount it.

 **Note:**

When an instance is deleted and re-created or shut down and restarted, storage volumes that were attached manually (that is, not attached automatically through the orchestration that was used to create the instance) must be attached again.

To prevent the boot issue, either do not add entries for manually attached volumes or use the 'nofail' option and set the last field to zero (don't fsck) in `/etc/fstab` for any manually attached volume, such as:

```
/dev/xvdd /mnt/store ext3 defaults,nofail 0 0
```

1. Connect to the instance using `ssh`. See [Log In to a VM Using SSH](#).

2. List the devices available on your instance:

```
ls /dev/xvd*
```

Device names start from `/dev/xvdb` and are determined by the index number that you assigned when you attached the storage volumes. For example, if you attached a storage volume at index 1, the volume gets the device name, `/dev/xvdb`. The storage volume at index 2 would be `/dev/xvdc`, the storage volume at index 3 would be `/dev/xvdd`, and so on.

3. Identify the device name corresponding to the disk number that you want to mount.

For example, if you want to mount the storage volume that you had attached at index 3, the device name would be `/dev/xvdd`.

4. When mounting a storage volume for the first time, after formatting the storage volume, use a tool such as `mkfs` to create a file system on the storage volume. For example, to create an `ext3` file system on `/dev/xvdd`, run the following command:

```
sudo mkfs -t ext3 /dev/xvdd
```

 **Note:**

If the Extended File System utilities aren't available on your instance, a message such as the following is displayed:

```
mkfs.ext3: No such file or directory
```

To install the Extended File System utilities, run the following command:

```
sudo yum install e4fsprogs
```

5. Create a mount point on your instance. For example, to create the mount point `/mnt/store`, run the following command:

```
sudo mkdir /mnt/store
```

6. Mount the storage volume on the mount point that you created on your instance. For example, to mount the device `/dev/xvdd` at the `/mnt/store` directory, run the following command:

```
sudo mount /dev/xvdd /mnt/store
```

If you prefer, you can specify the disk UUID instead of the device name in the `mount` command. To find out the UUID of the disks attached to your instance, run the `blkid` command.

7. To make the mount persistent across instance restarts, edit the `/etc/fstab` file and add the mount as an entry in that file.

 **Note:**

When an instance is deleted and re-created, or shut down and restarted, storage volumes that were attached manually (that is, not attached automatically through the orchestration used to create the instance) are not attached automatically. To prevent the boot issue, either do not add entries for manually attached volumes or use the 'nofail' option and set the last field to zero (don't fsck) in `/etc/fstab` for any manually attached volume, such as:

```
/dev/xvdd /mnt/store ext3 defaults,nofail 0 0
```

For information about unmounting a storage volume, see [Unmounting a Storage Volume from a Linux Instance](#) in *Using Oracle Cloud Infrastructure Compute Classic*.

Retrieve Predefined Instance Metadata

Two types of metadata are stored within your instances: *user-defined instance attributes* that you can define explicitly while creating instances, and *predefined instance metadata* fields that are stored by default for all instances.

Scripts and applications running on the instances can use the available metadata to perform certain tasks. For example, SSH public keys that are specified while creating an instance are stored as metadata on the instance. A script running on the instance can retrieve these keys

and append them to the `authorized_keys` file of specified users to allow key-based login to the instance using `ssh`.

1. Log in to the instance.
See [Log In to a VM Using SSH](#).
2. Get a list of the available metadata versions by running the following command:

```
curl http://192.0.0.192
```

3. From the list of versions displayed, select the version that you want to use.
4. Get a list of the top-level metadata fields:

```
curl http://192.0.0.192/{version}/meta-data
```

In this command, replace `{version}` with the version that you identified in the previous step.

Example:

```
curl http://192.0.0.192/2007-08-29/meta-data
```

5. Retrieve the specific metadata that you want, by running one of the following command examples:

 **Note:**

When you run these commands, replace `2007-08-29` with the metadata version that you want to use.

- To retrieve the private IP address of the instance:

```
curl http://192.0.0.192/2007-08-29/meta-data/local-ipv4
10.106.15.70
```

- To retrieve the host name of the instance:

```
curl http://192.0.0.192/2007-08-29/meta-data/local-hostname
bd6032.acme.oraclecloud.com
```

- To retrieve information about the memory and CPU resources of the instance:

```
curl http://192.0.0.192/2007-08-29/meta-data/instance-type
7680 ram, 2.0 cpus
```

- To retrieve the instance name:

```
curl http://192.0.0.192/2007-08-29/meta-data/instance-id
/Compute-acme/joe.jonathan@example.com/4c318760-444b-4b48-83e1-
e1b112c201f2
```

- To find out how many SSH public keys are stored on the instance:

```
curl http://192.0.0.192/2007-08-29/meta-data/public-keys
0
1
2
```

In this example, three SSH public keys are stored as metadata, with index numbers 0, 1, and 2.


- To retrieve the value of a specific SSH public key:

```
curl http://192.0.0.192/2007-08-29/meta-data/public-keys/0/openssh-key  
ssh-rsa AAAAB3NzaC1yc2EAAAABI... == joe.jonathan@acme.com
```

Delete and Re-create an Instance


After creating an instance, if you no longer need the instance, you can delete it. If you want to use the same instance again later on, you can re-create the instance.

To learn what happens when you delete and restart an instance, see *Deleting and Re-creating an Instance* in *Using Oracle Cloud Infrastructure Compute Classic*.

1. Sign in to the Compute Classic console.
2. On the Instances page, identify the instance that you want to delete.
3. Click the **Orchestrations** tab.
4. Go to the orchestration that controls the instance that you want to delete.
5. From the  menu, select **Suspend**. The status of the orchestration changes to **Suspending**. After all nonpersistent objects have been deleted, the status of the orchestration changes to **Suspended**.

Caution:

If you terminate the orchestration instead of suspending it, all the objects created by the orchestration are deleted, including persistent objects such as storage volumes.

6. When you are ready to re-create the instance, on the Orchestrations page, go to the orchestration that controls the instance that you want to re-create. From the  menu, select **Start**.

The status of the orchestration changes to **Starting**. After all objects have been created, the status of the orchestration changes to **Ready**.

10

Compute Classic: Advanced Tasks

Topics

- [Control Network Traffic](#)
- [Create a Bootable Volume](#)
- [Create an Instance Snapshot](#)
- [Register a Machine Image](#)
- [Create a Colocated Volume Snapshot](#)
- [Restore a Volume from a Snapshot](#)
- [Create Resources Using an Orchestration](#)
- [Create a Multi-Tier Topology with IP Networks Using an Orchestration](#)
- [Manage Resources Using Terraform](#)
- [Create a Multi-Tier Topology with IP Networks Using Terraform](#)

Control Network Traffic

The steps to control network traffic for your VMs vary depending on whether the VMs are attached to the shared network or to IP networks.

Scenario 1: Open Ports for VMs Attached to the Shared network

Network traffic to and from VMs attached to the shared network is controlled by security rules that you define and also the access policy defined for the security list that the VMs are in. By default, the outbound policy of a security list is permit and the inbound policy is deny. To permit traffic to the VMs, you must create the necessary security rules.

Prerequisites

1. Identify the source and destination security lists that contain the VMs for which you want to open ports. If you want to use new security lists, then create them and add your VMs to the security lists. See [Creating a Security List and Adding an Instance to a Security List](#) in *Using Oracle Cloud Infrastructure Compute Classic*.
2. Identify the protocol for which you want to allow traffic. Note that for well known protocols such as HTTPS (port 443), SSH (port 22), and ICMP (for ping requests), Oracle provides predefined protocols (called security applications) that you can use in your security rules.

If you want to create a security application, then complete the steps in [Creating a Security Application](#) in *Using Oracle Cloud Infrastructure Compute Classic*.
3. (Optional) If the source or destination for which you want to permit network traffic is a specific set of hosts outside the Compute Classic site, then create the required security IP lists as described in [Creating a Security IP List](#) in *Using Oracle Cloud Infrastructure Compute Classic*.

Procedure

Create a security rule to permit traffic to the VM.

1. Sign in to the Compute Classic console.
2. Click the **Network** tab.
3. Expand **Shared Network** in the left navigation pane, and then click **Security Rules**.
4. Click **Create Security Rule**.
5. In the **Create Security Rule** dialog box, select or enter the following information:
 - **Name:** Enter a name for the security rule.
 - **Status:** Select **Enabled**.
 - **Security Application:** Select the security application that you identified (or created) earlier.
 - **Source**
 - If the source from which you want to permit network traffic is a security list within the site, then select that security list.
 - If the source from which you want to permit network traffic is a set of hosts outside the site or the public Internet, then select the appropriate security IP list.
 - **Destination**
 - If the destination to which you want to permit network traffic is a security list within the site, then select that security list.
 - If the destination to which you want to permit network traffic is a set of hosts outside the site or the public Internet, then select the appropriate security IP list.
 - **Description:** Enter a meaningful description for the new rule.
6. Click **Create**.

Scenario 2: Permit Network Traffic for VMs Attached to IP networks

Network traffic to and from a VM attached to an IP network is controlled by access control lists (ACLs) containing security rules that you define and apply to the appropriate vNICsets.

Prerequisites

- Identify the source and destination vNICsets that contain to the vNICs for which you want to control network access. To create a new vNICset, complete the steps in *Creating a vNICset in Using Oracle Cloud Infrastructure Compute Classic*.
- (Optional) If the source or destination for which you want to permit network traffic is a specific set of hosts outside the Compute Classic site, then create the required IP address prefix sets as described in *Creating an IP Address Prefix Set in Using Oracle Cloud Infrastructure Compute Classic*.
- Identify the ACL in which you want to define the required security rules. If you want to use a new ACL, then create it as described in *Creating an ACL in Using Oracle Cloud Infrastructure Compute Classic*.
- Identify the security protocols for which you want to define security rules. Note that for well known protocols such as HTTPS (port 443), SSH (port 22), and ICMP (for

ping requests), Oracle provides predefined security protocols that you can use in your security rules.

If you want to create a security protocol, then complete the steps in *Creating a Security Protocol for IP Networks* in *Using Oracle Cloud Infrastructure Compute Classic*.

Procedure

Create an ingress security rule to permit traffic to the VM.

1. Sign in to the Compute Classic console.
2. Click the **Network** tab.
3. Expand **IP Network** in the left navigation pane, and then click **Security Rules**.
4. Click **Create Security Rule**.
5. In the **Create Security Rule** dialog box, select or enter the following information:
 - **Name:** Enter a name for the security rule.
 - **Status:** Select **Enabled**.
 - **Type:** Select **Ingress** or **Egress**, as appropriate.
 - **Access Control List:** Select the ACL that you identified (or created) earlier.
 - **Security Protocols:** Select the security protocols that you identified (or created) earlier.
 - **Source IP Address Prefix Sets:** If the source from which you want to permit network traffic is a set of hosts outside the site, then select the appropriate IP address prefix set that you created earlier.
 - **Source vNICset:** If the source from which you want to permit network traffic is a vNICset within the site, then select the vNICset that you identified or created earlier.
 - **Destination IP Address Prefix Sets:** If the destination to which you want to permit network traffic is a set of hosts outside the site, then select the appropriate IP address prefix set that you created earlier.
 - **Destination vNICset:** If the destination to which you want to permit network traffic is a vNICset within the site, then select the vNICset that you identified or created earlier.
 - **Description:** Enter a meaningful description for the new rule.
 - **Tags:** Select the tags to be assigned to the rule.
6. Click **Create**.

Scenario 3: Control Network Traffic for VMs Attached to the Shared Network and to IP Networks

Complete the steps for scenario 1 and scenario 2.

Create a Bootable Volume

A **storage volume** is a virtual disk that provides persistent block storage space for instances in Compute Classic. While creating a storage volume, you can associate it with a machine image and later use this storage volume as the boot disk for an instance. When you boot an instance from such a storage volume, any changes you make to the boot disk aren't lost when the instance is deleted and re-created..

1. Sign in to the Compute Classic console.

2. Click the **Storage** tab.
3. Click **Create Storage Volume**.
4. Select or enter the required information:
 - Enter a name for the storage volume. Note this name. You'll need it later to search for the storage volume on the Storage page.


Pick a name that you can use later to quickly identify the key characteristics of the storage volume. For example, consider a name such as `boot-OL66-20G` for a bootable storage volume with an Oracle Linux 6.6 machine image on a 20-GB disk).
 - Select a machine image in the **Boot Image** field.

If you select a machine image with a large disk size, it may take a while for the storage volume to be created.
 - The size of the volume is set automatically based on the image you selected. Leave it as is, or enter a larger size.
 - Leave the other fields at the default values.
5. Click **Create**.

The Storage page is displayed.

While the new storage volume is being created, the **Status** field for the storage volume shows **Initializing**.

When the storage volume is ready, the **Status** field changes to **Online**. You can then specify this storage volume as the boot disk while creating an instance.

To view details of the new storage volume, search for it using the name you noted earlier. From the  menu, select **View**.

Create an Instance Snapshot


Creating a snapshot of an instance allows you to capture the current state of the nonpersistent boot disk used by an instance, including all customization that you may have made at the operating-system level after creating the instance.



Note:

Instance snapshots capture the state of your nonpersistent boot disk.

You can't create an instance snapshot if your instance uses a persistent boot disk. For the steps to take a snapshot of a bootable volume, see [Create a Colocated Volume Snapshot](#).

1. Sign in to the Compute Classic console.
2. Locate the instance that you want to create a snapshot of. From the  menu, select **Create Snapshot**.
3. In the Create Instance Snapshot dialog box, enter a name for the snapshot.

4. If you haven't yet finished customizing your instance and you want to create the snapshot just before you delete the instance, you can select the **Deferred Snapshot** option. This option allows you to continue working on the instance. The snapshot is taken only when you delete the instance or stop the instance orchestration.
5. Click **Create**. A request to create an instance snapshot is created. If the deferred snapshot option was selected, the snapshot will be generated when you delete the instance. If the deferred snapshot option wasn't selected, the process of creating the instance snapshot begins right away.

When an instance snapshot is generated, it creates a custom image. While the image is being created, or when you select the option to create a deferred snapshot, the instance details page shows the state of the instance snapshot as *Active*. When the image has been created and is available in your Oracle Cloud Infrastructure Object Storage Classic account, the state of the instance snapshot changes to *Complete*.

Next step: Register the snapshot as an image. See [Register a Machine Image](#).


Register a Machine Image

An instance snapshot captures the current state of the nonpersistent boot disk of an instance and uses it to create a corresponding machine image. You can then use this machine image to create other instances. These instances are clones of the instance that you created the snapshot of. Any customization done on that instance is automatically part of instances created using the snapshot.

The image created by an instance snapshot is stored in Object Storage. Before you can use this image to create an instance, you must register this image.

1. Sign in to the Compute Classic console.
2. Click the **Instance Snapshots** tab in the left pane.

The Instance Snapshots page displays a list of snapshots. Instance snapshots are listed alphabetically by instance name. If an instance has multiple snapshots, the most recent snapshot is listed first.


3. Go to the snapshot that you want to use. From the  menu, select **Associate Image**.
4. Enter a description for the image and click **Ok**.

After you register the machine image, it is available as a private image that you can create instances from.

Create a Colocated Volume Snapshot

Creating a snapshot of a storage volume enables you to capture all the data that is currently stored on the storage volume.

If the storage volume is attached to an instance, then only data that has already been written to the storage volume will be captured in the snapshot. Data that is cached by the application or the operating system will be excluded from the snapshot.


1. Sign in to the Compute Classic console.
2. Click the **Storage** tab.
3. Go to the storage volume that you want to create a snapshot of. From the  menu, select **Create Snapshot**.

4. In the Create Storage Snapshot dialog box, enter the required information:
 - **Name:** Enter a name for the snapshot.
 - **Colocated:** Select this check box.
 - **Description:** Enter a description for the snapshot.
 - **Tags:** Enter tags to help identify your snapshot, if required.
5. Click **Create**.
A storage volume snapshot is generated.
6. To view the available snapshots, click the **Storage Volumes** drop-down list, and select **Storage Snapshots**.

After creating a volume snapshot, you can use it to create a storage volume that's identical to the original volume. See [Restore a Volume from a Snapshot](#).

Restore a Volume from a Snapshot

After taking a snapshot of a volume, you can use the snapshot to create a new volume that's identical to the original.

1. Sign in to the Compute Classic console.
2. Click the **Storage** tab.
3. From the **Storage Volumes** drop-down list, select **Storage Snapshots**.
4. Locate the snapshot from which you want to create a volume. From the  menu, select **Restore Volume**.
5. In the Restore Storage Volume dialog box, enter a name for the new storage volume and specify a description, if required.
6. Click **Restore**.

While the storage volume is being created, the status on the Storage Volume page is **Initializing**. After the storage volume is created, the status changes to **Online**.

Create Resources Using an Orchestration

You can create a blank orchestration, and then add objects to it by updating the orchestration. While updating the orchestration, you can define attributes for a single instance or create complex topologies that consist of multiple instances and multiple networks.

Scenario Overview

In this example, you create the following resources:

- An IP network
- A vNICset
- A VM attached to the IP network
- An SSH public key associated with the VM
- A storage volume attached to the VM
- A public IP reservation for the VM

- A security protocol for SSH traffic to the VM
- A security rule to permit SSH access to the VM, and an ACL for the security rule

Prerequisite



Generate an SSH key pair. In the orchestration, you'll add the public key and associate it with the instance. See [Generate an SSH Key Pair](#).

Procedure





Note:


This procedure walks you through the key steps required to quickly provision the basic compute and networking resources. It does not cover the advanced configuration options.

1. Sign in to the Compute Classic console.
2. Click the **Orchestrations** tab.
3. Click **Create Orchestration**.
The Create Orchestration dialog box appears.
4. In the Create Orchestration dialog box, enter the following information.
 - **Name:** Enter a name for the orchestration.
 - **Description:** Enter a description.
 - **Tags:** Specify one or more tags to help you identify and categorize the orchestration.
5. Click **Create**.
A blank orchestration is created and listed on the Orchestrations page. You can now add objects by updating the orchestration.
6. From the  menu for the orchestration that you created, select **Update**.
The Orchestration page has a **JSON** section that shows the current orchestration definition. As you add and update the objects in the orchestration, the JSON section gets updated. Note that the objects you add and update are in the **Inactive** status. They are created only when you start the orchestration.
In this example, you create an Oracle Linux instance attached to an IP network and accessible over the public Internet by using SSH.
7. Add an ACL.
 - a. Expand the **Access Control List** section, and click **Add**.
 - b. Enter a name for the ACL.
Note this name. You'll need to specify it later when configuring the security rule to permit SSH access.
 - c. Click **Create**.
 - d. From the  menu for the ACL that you added, select **Properties**.
 - e. In the Object Properties dialog box, select the **Persistent** check box.
 - f. Click **Update**.





8. Add an IP network.
 - a. Expand the **IP Network** section, and click **Add**.
 - b. In the **Name** field, enter a name for the IP network.

Note this name. You'll need to specify it later when configuring the network interface of your instance.
 - c. In the **IP Address Prefix** field, enter the address range of the IP network in the CIDR format (example: 192.168.10.0/28).
 - d. Click **Create**.
 - e. From the  menu for the IP network that you added, select **Properties**.
 - f. In the Object Properties dialog box, select the **Persistent** check box.
 - g. Click **Update**.
9. Add an IP reservation.
 - a. Expand the **IP Reservation (IP Network)** section, and click **Add**.
 - b. In the Create IP Reservation dialog box, enter a name for the IP reservation.

Note this name. You'll need to specify it later when configuring the network interface of your instance.
 - c. Click **Create**.
 - d. From the  menu for the IP reservation that you added, select **Properties**.
 - e. In the Object Properties dialog box, select the **Persistent** check box.
 - f. Click **Update**.
10. Add a security protocol for SSH traffic.
 - a. Expand the **Security Protocol** section, and click **Add**.
 - b. In the Create Security Protocol dialog box, provide the following information:
 - **Name**: Enter a name for the protocol.

Note this name. You'll need to specify it later when configuring the security rule to permit SSH access.
 - **IP Protocol**: Select **TCP**.
 - **Destination Port Set**: Enter **22**.
 - c. Click **Create**.
 - d. From the  menu for the protocol that you added, select **Properties**.
 - e. In the Object Properties dialog box, select the **Persistent** check box.
 - f. Click **Update**.
11. Add the SSH public key.
 - a. Expand the **SSH Key** section, and click **Add**.
 - b. Enter a name for the key.

Note this name. You'll need to specify it later when selecting the public key for your instance.
 - c. Click **Select File**.

- d. Browse to the file that contains the public key you generated earlier, and select it.
 - e. Click **Add**.
 - f. From the  menu for the SSH key that you added, select **Properties**.
 - g. In the Object Properties dialog box, select the **Persistent** check box.
 - h. Click **Update**.
12. Add a vNICset.
- a. Expand the **Virtual NIC Set** section, and click **Add**.
 - b. Enter a name for the vNICset.
Note this name. You'll need to specify it later when configuring the network interface of your instance.
 - c. Click **Create**.
 - d. From the  menu for the vNICset that you added, select **Properties**.
 - e. In the Object Properties dialog box, select the **Persistent** check box.
 - f. Click **Update**.
13. Add a security rule to permit SSH access to the instance.
- a. Expand the **Security Rule (IP Network)** section, and click **Add**.
 - b. In the Create Security Rule dialog box, provide the following information:
 - **Name**: Enter a name for the security rule.
 - **Access Control List**: Select the ACL that you created.
 - **Security Protocols**: Select the SSH protocol that you created.
 - **Destination vNICset**: Select the vNICset that you created.Leave all the other fields at the default values.
 - c. Click **Create**.
 - d. From the  menu for the security rule that you added, select **Properties**.
 - e. In the Object Properties dialog box, select the **Persistent** check box.
 - f. Click **Update**.
14. Add a storage volume.
- a. Expand the **Storage Volume** section, and click **Add**.
 - b. In the Create Storage Volume dialog box, provide the following information:
 - **Name**: Enter a name for the volume.
 - **Size**: Set the required size.Leave all the other fields at the default values.
 - c. Click **Create**.
15. Add an instance, and configure networking for it.
- a. Expand the **Instance** section, and click **Add**.
 - b. From the  menu for the instance that you added, select **Update**.

- c. In the **IP Network Interfaces** section, click **Add IP Network Interface**, and provide the following information:
 - **IP Network:** Select the IP network that you added.
 - **Public IP Address:** Select the IP reservation that you added.
 - **Virtual NIC Sets:** Remove the **default** vNICset, and select the vNICset that you added.
 - d. In the **SSH Public Keys** section, click **Add SSH Public Key**, and add the public key that you uploaded.
 - e. In the **Storage Volumes** section, click **Attach Storage Volume**.
 - f. Select the volume that you created, and click **Attach**.
16. Scroll to the top of the page, and click **Back to Orchestration Details**.
 17. Click the **Start** button, near the upper-right corner.

When you start the orchestration, the status of the orchestration changes to **Starting** and then to **Ready** when all the objects defined in the orchestration are created successfully. The instance and other objects are created and their status changes from **Inactive** to **Active**.

18. At the confirmation prompt, click **Yes**.

In the **Information** pane at the top, the **Status** field shows **Starting**.

Wait until the status changes to **Ready**. Periodically, click the refresh button near the upper-right corner of the Information pane.

19. Verify that all the resources are created.

In all the resource sections, the status field shows **Active**.

You have successfully created your instance and the networking resources required to enable SSH connections to the instance.

Create a Multi-Tier Topology with IP Networks Using an Orchestration

Create an orchestration to launch and manage a multi-tier topology for an application deployed on Compute Classic instances attached to IP networks.

Topics

- [Scenario Overview](#)
- [Create the Orchestration](#)
- [\(Optional\) Verify Network Access to the VMs](#)

Scenario Overview

The application and the database that the application uses are hosted on instances attached to separate IP networks. Users outside Oracle Cloud have HTTPS access to the application instances. The topology also includes an admin instance that users outside the cloud can connect to using SSH. The admin instance can communicate with all the other instances in the topology.

 **Note:**

The focus of this guide is the network configuration for instances attached to IP networks in a sample topology. The framework and the flow of the steps can be applied to other similar or more complex topologies. The steps for provisioning or configuring other resources (like storage) are not covered in this guide.

Compute Topology

The topology that you are going to build using the steps in this tutorial contains the following Compute Classic instances:

- Two instances – `appVM1` and `appVM2` – for hosting a business application, attached to an IP network, each with a fixed public IP address.
- Two instances – `dbVM1` and `dbVM2` – for hosting the database for the application. These instances are attached to a second IP network.
- An admin instance – `adminVM` – that's attached to a third IP network and has a fixed public IP address.

 **Note:**

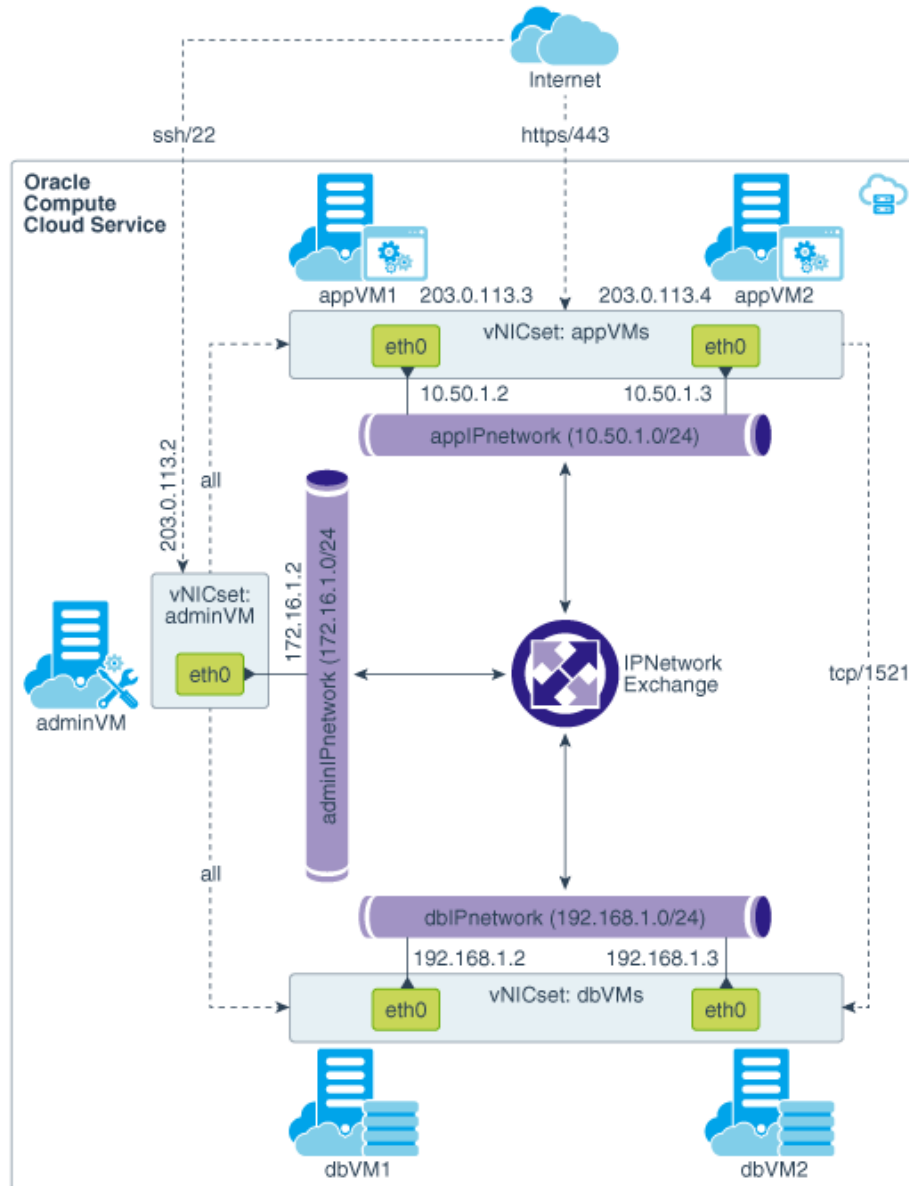
You won't actually install any application or database. Instead, you'll simulate listeners on the required application and database ports using the `nc` utility. The goal of this section is to demonstrate the steps to configure the networking that's necessary for the traffic flow requirements described next.

Traffic Flow Requirements

Only the following traffic flows must be permitted in the topology that you'll build:

- HTTPS requests from anywhere to the application instances
- SSH connections from anywhere to the admin instance
- All traffic from the admin instance to the application instances
- All traffic from the admin instance to the database instances
- TCP traffic from two application instances to port 1521 of the database instances

Topology Architecture Diagram



Network Resources Required for this Topology

- **Public IP address reservations** for the application instances and for the admin instance
- Three **IP networks**, one each for the application instances, the database instances, and the admin instance
- An **IP network exchange** to connect the IP networks in the topology
- **Security protocols** for SSH, HTTPS, and TCP/1521 traffic
- **ACLs** that will contain the required security rules
- **vNICsets** for the application instances, database instances, and the admin instance

- **Security rules** to allow SSH connections to the admin instance, HTTPS traffic to the application instances, and TCP/1521 traffic to the database instances

Create the Orchestration

You can create a blank orchestration, and then add objects to it by updating the orchestration. While updating the orchestration, you can define attributes for a topology that consists of multiple instances and multiple networks.

Prerequisite

Generate an SSH key pair. In the orchestration, you'll add the public key and associate it with the instance. See [Generate an SSH Key Pair](#).

Procedure



Note:

This procedure walks you through the key steps required to quickly provision the basic compute and networking resources. It does not cover the advanced configuration options.

1. Sign in to the Compute Classic console.
2. Go to the **Orchestrations** tab, and click **Create Orchestration**.
3. In the Create Orchestration dialog box, enter the following information.
 - **Name:** Enter a name for the orchestration.
 - **Description:** Enter a description.
 - **Tags:** Specify one or more tags to help you identify and categorize the orchestration.

Click **Create**.

A blank orchestration is created and listed on the Orchestrations page. You can now add objects by updating the orchestration.

4. From the  menu for the orchestration that you created, select **Update**.

The Orchestration page has a **JSON** section that shows the current orchestration definition. As you add and update the objects in the orchestration, the JSON section gets updated. Note that the objects you add and update are in the **Inactive** status. They are created only when you start the orchestration.



5. Add the following access control lists:

Purpose	Suggested Name
For the admin VM	adminVM
For the application VMs	appVMs
For the database VMs	dbVMs

Add the ACLs, one at a time, using the following steps:


- a. In the **Access Control List** section, and click **Add**.
- b. Enter a name for the ACL, as suggested in the table..

Note this name. You'll need to specify it later when configuring the security rules.

- c. Click **Create**.
 - d. From the  menu for the ACL that you added, select **Properties**.
 - e. In the Object Properties dialog box, select the **Persistent** check box.
 - f. Click **Update**.
6. Add an IP exchange.
 - a. In the **IP Exchange** section, and click **Add**.
 - b. In the Create IP Exchange dialog box, enter a name for the IP exchange.
Note this name. You'll need to specify it later when you add the IP networks.
 - c. Click **Create**.
 - d. From the  menu for the IP reservation that you added, select **Properties**.
 - e. In the Object Properties dialog box, select the **Persistent** check box.
 - f. Click **Update**.
 7. Add the following IP networks:

Purpose	Suggested Name	Suggested IP Address Prefix
For the admin VM	adminIPnetwork	172.16.1.0/24
For the application VMs	appIPnetwork	10.50.1.0/24
For the database VMs	dbIPnetwork	192.168.1.0/24


Add the vNICsets, one at a time, using the following steps:

- a. In the **IP Network** section, and click **Add**.
 - b. In the **Name** field, enter a name for the IP network, as suggested in the table..
Note this name. You'll need to specify it later when configuring the network interface of your instance.
 - c. In the **IP Address Prefix** field, enter the address range of the IP network in the CIDR format.
 - d. In the **IP Exchange** field, select the IP exchange that you created earlier.
 - e. Click **Create**.
 - f. From the  menu for the IP network that you added, select **Properties**.
 - g. In the Object Properties dialog box, select the **Persistent** check box.
 - h. Click **Update**.
8. Add the following IP reservations:

For VM	Suggested Name
Admin VM	ipResForAdminVM
Application VM 1	ipResForAppVM1
Application VM 2	ipResForAppVM2


Add the IP reservations, one at a time, using the following steps:

- a. In the **IP Reservation (IP Network)** section, and click **Add**.
 - b. In the Create IP Reservation dialog box, enter a name for the IP reservation, as suggested in the table..

Note this name. You'll need to specify it later when configuring the network interface of your instance.
 - c. Click **Create**.
 - d. From the  menu for the IP reservation that you added, select **Properties**.
 - e. In the Object Properties dialog box, select the **Persistent** check box.
 - f. Click **Update**.
9. Add the following security protocols:

Purpose	Suggested Name	IP Protocol	Destination Port Set
For HTTPS requests to the application VMs	https	TCP	443
For SSH traffic	ssh	TCP	22
For TCP traffic from the application VMs to the database VMs	tcp1521	TCP	1521

Add the security protocols, one at a time, using the following steps:


- a. In the **Security Protocol** section, and click **Add**.
 - b. In the Create Security Protocol dialog box, provide the following information:
 - **Name:** Enter a name for the protocol, as suggested in the table.. Note this name. You'll need to specify it later when configuring the security rule to permit SSH access.
 - **IP Protocol:** Select **TCP**.
 - **Destination Port Set:** Enter the required port.
 - c. Click **Create**.
 - d. From the  menu for the protocol that you added, select **Properties**.
 - e. In the Object Properties dialog box, select the **Persistent** check box.
 - f. Click **Update**.
10. Add the following vNICsets:

Purpose	Suggested Name	Applied Access Control Lists
For the admin VM	adminVM	adminVM
For the application VMs	appVMs	appVMs
For the database VMs	dbVMs	dbVMs

Add the vNICsets, one at a time, using the following steps:

- a. In the **Virtual NIC Set** section, and click **Add**.
- b. Enter a name for the vNICset, as suggested in the table.

Note this name. You'll need to specify it later when configuring the network interface of your instance.

- c. In the **Applied Access Control Lists** field, select the ACL as specified in the table.
 - d. Click **Create**.
 - e. From the  menu for the vNICset that you added, select **Properties**.
 - f. In the Object Properties dialog box, select the **Persistent** check box.
 - g. Click **Update**.
11. Add the following security rules:



Purpose	Suggested Name	Type	ACL	Source and Destination	Protocol
SSH requests from any source to the admin VM	internet-to-adminVM	Ingress	adminVM	Source: Any Destination: adminVM vNICset	ssh
All traffic from the admin VM to any destination	adminVM-to-any	Egress	adminVM	Source: adminVM vNICset Destination: Any	Any
All traffic from the admin VM to the application VMs	adminVM-to-appVMs	Ingress	appVMs	Source: adminVM vNICset Destination: appVMs vNICset	Any
HTTPS traffic from any source to port 443 of the application VMs	internet-to-appVMs	Ingress	appVMs	Source: Any Destination: appVMs vNICset	https
TCP traffic from the application VMs to port 1521 of the DB VMs	appVMs-to-dbVMs-egress	Egress	appVMs	Source: appVMs vNICset Destination: dbVMs vNICset	tcp1521
TCP traffic from the application VMs to port 1521 of the DB VMs	appVMs-to-dbVMs-ingress	Ingress	dbVMs	Source: appVMs vNICset Destination: dbVMs vNICset	tcp1521
All traffic from the admin VM to the DB VMs	adminVM-to-dbVMs	Ingress	dbVMs	Source: adminVM vNICset Destination: dbVMs vNICset	Any

Add the security rules, one at a time, using the following steps:

- a. Expand the **Security Rule (IP Network)** section, and click **Add**.
- b. In the Create Security Rule dialog box, provide the following information:
 - **Name:** Enter a name for the security rule, as suggested in the table..
 - **Type:** Select **Ingress** or **Egress**, as specified in the table.
 - **Access Control List:** Select the ACL specified in the table.
 - **Security Protocols:** Select the SSH protocol specified in the table. If the table shows **Any**, then leave this field blank.


- **Source vNICset:** Select the source specified in the table. If the table shows **Any**, then leave this field at **Not Set**.
- **Destination vNICset:** Select the destination specified in the table. If the table shows **Any**, then leave this field at **Not Set**.

Leave all the other fields at the default values.

- Click **Create**.
 - From the  menu for the security rule that you added, select **Properties**.
 - In the Object Properties dialog box, select the **Persistent** check box.
 - Click **Update**.
12. Add the SSH public key.
- Expand the **SSH Key** section, and click **Add**.
 - Enter a name for the key.
Note this name. You'll need to specify it later when selecting the public key for your instance.
 - Click **Select File**.
 - Browse to the file that contains the public key you generated earlier, and select it.
 - Click **Add**.
 - From the  menu for the SSH key that you added, select **Properties**.
 - In the Object Properties dialog box, select the **Persistent** check box.
 - Click **Update**.
13. Add the following VMs, and configure networking for them.

VM Name	Suggested DNS Hostname Prefix	Suggested vNIC Name	IP Network	Public IP Address	Virtual NIC Set
adminVM	adminvm	admn	adminIPnetwork	ipResForAdminVM	adminVM
appVM1	appvm1	app1	appIPnetwork	ipResForAppVM1	appVMs
appVM2	appvm2	app2	appIPnetwork	ipResForAppVM2	appVMs
dbVM1	dbvm1	db1	dbIPnetwork	None	dbVMs
dbVM2	dbvm2	db2	dbIPnetwork	None	dbVMs

Add the VMs, one at a time, using the following steps:

- In the **Instance** section, and click **Add**.
- From the  menu for the instance that you added, select **Update**.
- In the **Information** section, complete the following steps:
 - **Name:** Enter a name that you can use to easily identify the VM, as suggested in the table.
 - **Image:** Select an image of your choice.

 **Note:**

The optional steps at the end to verify network access are for VMs created using Oracle Linux 6.8 and 7.2 images. Those optional steps might not work for VMs created using other images.

DNS Hostname Prefix: Enter a host name as suggested in the table.

- If you want to, change the shape and other settings of the VM as required. The default values will work for this example.
- d. Click **Update**.
 - e. In the **IP Network Interfaces** section, click **Add IP Network Interface**, and provide the following information:
 - **vNIC Name:** Enter a unique name for the vNIC, as suggested in the table.app
 - **IP Network:** Select the IP network specified in the table.
 - **Public IP Address:** Select the IP reservation specified in the table.
 - **Virtual NIC Sets:** Remove the **default** vNICset, and select the vNICset that you added.
 - f. Click **Save**.
 - g. In the **SSH Public Keys** section, click **Add SSH Public Key**, and add the public key that you uploaded.
 - h. Scroll to the top of the page, and click **Back to Orchestration Details**.
14. After adding all the VMs, scroll to the top of the page, and click the **Start** button near the upper-right corner.

When you start the orchestration, the status of the orchestration changes to **Starting** and then to **Ready** when all the objects defined in the orchestration are created successfully. The instance and other objects are created and their status changes from **Inactive** to **Active**.

15. At the confirmation prompt, click **Yes**.

In the **Information** pane at the top, the **Status** field shows **Starting**.

Wait until the status changes to **Ready**. Periodically, click the refresh button near the upper-right corner of the Information pane.

16. Verify that all the resources are created.

In all the resource sections, the status field shows **Active**.

You have successfully created your instances and the required networking resources.

(Optional) Verify Network Access to the VMs

Before You Begin

Identify the IP addresses of the VMs:

- adminVM: Public IP address
- appVM1: Public and private IP addresses
- appVM2: Public and private IP addresses
- dbVM1: Private IP address
- dbVM2: Private IP address

In the **Instances** tab of the web console, locate the VM, and note the addresses in the **Public IP** and **Private IP** columns.



Note:

If the **Public IP** column is blank for a VM, then on the **Network** tab, under **IP Network**, select **IP Reservations**, and note the public IP address shown there for the reservation that's assigned to the VM.

Verify SSH Connections from Outside the Cloud to the Admin VM

Run the following command from your local machine:

```
[localmachine ~]$ ssh -i path-to-privateKeyFile opc@publicIPAddressOfAdminVM
```

You should see the following prompt:

```
opc@adminvm
```

This confirms that SSH connections can be made from outside the cloud to the admin VM.

Verify SSH Connections from the Admin VM to the Database and Application VMs

1. Copy the private SSH key file corresponding to the public key that you associated with your VMs from your local machine to the admin VM, by running the following command on your local machine:

```
[localmachine ~]$ scp -i path-to-privateKeyFile path-to-privateKeyFile
opc@publicIPAddressOfAdminVM:~/.ssh/privatekey
```

2. From your local machine, connect to the admin VM using SSH:

```
[localmachine ~]$ ssh -i path-to-privateKeyFile
opc@publicIPAddressOfAdminVM
```

3. From the admin VM, connect to each of the database and application VMs using SSH:

```
[opc@adminvm]$ ssh -i ~/.ssh/privatekey opc@privateIPAddress
```

4. Depending on the VM you connect to, you should see one of the following prompts after the `ssh` connection is established.
 - `opc@appvm1`
 - `opc@appvm2`

- opc@dbvm1
- opc@dbvm2

Verify Connectivity from Outside the Cloud to Port 443 of the Application VMs

You can use the `nc` utility to simulate a listener on port 443 on one of the application VMs, and then run `nc` from any host outside the cloud to verify connectivity to the application VM.

Note:

The verification procedure described here is specific to VMs created using the Oracle-provided images for Oracle Linux 7.2 and 6.8.

1. On your local host, download the `nc` package from http://yum.oracle.com/repo/OracleLinux/OL6/latest/x86_64/getPackage/nc-1.84-24.el6.x86_64.rpm.
2. Copy `nc-1.84-24.el6.x86_64.rpm` from your local host to the admin VM.

```
[localmachine ~]$ scp -i path-to-privateKeyFile
path_to_nc-1.84-24.el6.x86_64.rpm opc@publicIPAddressOfAdminVM:~
```

3. From your local machine, connect to the admin VM using SSH:

```
[localmachine ~]$ ssh -i path-to-privateKeyFile
opc@publicIPAddressOfAdminVM
```

4. Copy `nc-1.84-24.el6.x86_64.rpm` to one of the application VMs.

```
[opc@adminvm]$ scp -i ~/.ssh/privatekey ~/nc-1.84-24.el6.x86_64.rpm
opc@privateIPAddressOfAppVM1:~
```

5. Connect to the application VM:

```
[opc@adminvm]$ ssh -i ~/.ssh/privatekey opc@privateIPAddressOfAppVM1
```

6. On the application VM, install `nc`.

```
[opc@appvm1]$ sudo rpm -i nc-1.84-24.el6.x86_64.rpm
```

7. Configure the application VM to listen on port 443. Note that this step is just for verifying connections to port 443. In real-life scenarios, this step would be done when you configure your application on the VM to listen on port 443.

```
[opc@appvm1]$ sudo nc -l 443
```

8. From any host outside the cloud, run the following `nc` command to test whether you can connect to port 443 of the application VM:

```
[localmachine ~]$ nc -v publicIPAddressOfAppVM1 443
```

The following message is displayed:

```
Connection to publicIPAddressOfAppVM1 443 port [tcp/https] succeeded!
```

This message confirms that the application VM accepts connection requests on port 443.

9. Press `Ctrl + C` to exit the `nc` process.

Verify Connectivity from the Application VMs to Port 1521 of the Database VMs

You can use the `nc` utility to simulate a listener on port 1521 on one of the database VMs, and then run `nc` from one of the application VMs to verify connectivity from the application tier to the database tier.

Note:

The verification procedure described here is specific to VMs created using the Oracle-provided images for Oracle Linux 7.2 and 6.8.

1. From your local machine, connect to the admin VM using SSH:

```
[localmachine ~]$ ssh -i path-to-privateKeyFile  
opc@publicIPAddressOfAdminVM
```

2. Copy `nc-1.84-24.el6.x86_64.rpm` to one of the database VMs.

```
[opc@adminvm]$ scp -i ~/.ssh/privatekey ~/nc-1.84-24.el6.x86_64.rpm  
opc@privateIPAddressOfDBVM1:~
```

3. Connect to the database VM:

```
[opc@adminvm]$ ssh -i ~/.ssh/privatekey opc@privateIPAddressOfDBVM1
```

4. On the database VM, install `nc`.

```
[opc@dbvm1]$ sudo rpm -i nc-1.84-24.el6.x86_64.rpm
```

5. Configure the VM to listen on port 1521. Note that this step is just for verifying connections to port 1521. In real-life scenarios, this step would be done when you set up your database to listen on port 1521.

```
[opc@dbvm1]$ nc -l 1521
```

6. Leave the current terminal session open. Using a new terminal session, connect to the admin VM using SSH and, from there, connect to one of the application VMs.

```
[localmachine ~]$ ssh -i path-to-privateKeyFile  
opc@publicIPAddressOfAdminVM  
[opc@adminvm]$ ssh -i ~/.ssh/privatekey opc@privateIPAddressOfAppVM1
```

7. From the application VM, run the following `nc` command to test whether you can connect to port 1521 of the database VM:

```
[opc@appvm1 ~]$ nc -v privateIPaddressOfDBVM1 1521
```

The following message is displayed:

```
Connection to privateIPaddressOfDBVM1 1521 port [tcp/ncube-lm]  
succeeded!
```

This message confirms that the database VM accepts connection requests received on port 1521 from the application VMs.

8. Press `Ctrl + C` to exit the `nc` process.

Manage Resources Using Terraform

Terraform is a third-party tool that you can use to create and manage your IaaS and PaaS resources on Oracle Cloud at Customer. This guide shows you how to install and configure Terraform, and then use it to deploy a sample set of Compute Classic resources.

Topics

- [Scenario Overview](#)
- [Prerequisites](#)
- [Create the Required Resources Using Terraform](#)
- [Add, Update, and Delete Resources Using Terraform](#)

Scenario Overview

In this example, you create the following Compute Classic resources:

- A persistent boot disk
- An IP network
- A vNICset
- A VM based on the image in the boot disk and attached to the IP network
- An SSH public key associated with the VM
- A data volume attached to the VM
- A public IP reservation for the VM
- A security protocol for SSH traffic to the VM
- A security rule to permit SSH access to the VM, and an ACL for the security rule

Prerequisites

1. If you are new to Terraform, learn the basics.

At a minimum, read the brief introduction here: <https://www.terraform.io/intro/index.html>.

2. Download and install Terraform on your local computer.

Binary packages are available for several operating systems and processor architectures. For the instructions to download and install Terraform, go to <https://www.terraform.io/intro/getting-started/install.html>.

3. Generate an SSH key pair. See [Generate an SSH Key Pair](#).

4. Gather the required Oracle Cloud account information:

- Your Oracle Cloud user name and password.
- The service instance ID.
 - a. Sign in to Oracle Cloud My Services.
 - b. Locate the Compute Classic tile and click **Compute Classic**.
 - c. Locate the **Service Instance ID** field, and note its value (example: 500099999).
- The REST endpoint URL for Compute Classic.
 - a. Sign in to Oracle Cloud My Services, using the My Services URL from the welcome email.
 - b. Click ☰ near the upper left corner of the page.
 - c. In the menu that appears, expand **Services**, and click **Compute Classic**. The Instances page of the Compute Classic web console is displayed.
 - d. Click **Site** near the top of the page, and select the site for which you want to find out the REST endpoint URL.
 - e. In the **Site Selector** dialog box, note the URL in the REST Endpoint field.

Create the Required Resources Using Terraform

Define the resources you need in a Terraform configuration and then apply the configuration.

1. On the computer where you installed Terraform, create a new directory.
2. In the new directory, create an empty text file, *name-of-your-choice.tf*.

This is a Terraform configuration. In this file, you define the following:

- The parameters that Terraform must use to connect to your Oracle Cloud at Customer machine
- The resources to be provisioned

! Important:

The `.tf` extension is mandatory. When Terraform performs any operation, it looks for a file with the `.tf` extension in the current directory.

3. Open the text file in an editor of your choice.

4. Add the following code to define the parameters that Terraform needs to connect to your account:

```
provider "opc" {  
  user          = "jack.smith@example.com"  
  password      = "mypassword"  
  identity_domain = "500099999"  
  endpoint      = "https://compute.site99.ocm.rack100.example.com"  
}
```

In this code:

- Don't change the `provider` line.
 - `user` and `password`: Replace with your Oracle Cloud credentials.
 - `identity_domain`: Replace with the service instance ID that you identified earlier.
 - `endpoint`: Replace with the REST endpoint URL of Compute Classic.
5. Add code for each resource that you want to create using Terraform.

 **Note:**

When copying and editing the code, follow the instructions carefully.

- a. Create an ACL by appending the following code.

```
# Create an ACL  
resource "opc_compute_acl" "default" {  
  name = "occACL"  
}
```

In this code:

- Don't change the `resource` line.
 - `name`: Replace with a name of your choice, or leave the example as is.
- b. Create an IP network by appending the following code:

```
# Create an IP network  
resource "opc_compute_ip_network" "default" {  
  name              = "occIPnetwork"  
  ip_address_prefix = "192.168.100.0/24"  
}
```

In this code:

- Don't change the `resource` line.
- `name`: Replace with a name of your choice, or leave the example as is.
- `ip_address_prefix`: Replace with an address range of your choice in CIDR format, or leave the value in the example as is.

- c. Reserve a public IP address for the VM by appending the following code:

```
# Reserve a public IP address
resource "opc_compute_ip_address_reservation" "default" {
  name          = "occIPreservation"
  ip_address_pool = "public-ippool"
  lifecycle {
    prevent_destroy = true
  }
}
```

In this code:

- Don't change the `resource` line.
- `name`: Replace with a name of your choice, or leave the example as is.
- `ip_address_pool`: You need a publicly routable IP address. So don't change this line.
- `lifecycle.prevent_destroy=true` reduces the chance of accidentally deleting the resource. This setting is useful for resources that you want to retain for future use even after you delete the VM.

- d. Define a security protocol for SSH by appending the following code:

```
# Create a security protocol for SSH
resource "opc_compute_security_protocol" "default" {
  name          = "occSSHprotocol"
  dst_ports    = ["22"]
  ip_protocol  = "tcp"
}
```

In this code:

- Don't change the `resource` line.
- `name`: Replace with a name of your choice, or leave the example as is.
- `dst_ports`: 22 is the port for the SSH protocol. So don't change this line.
- `ip_protocol`: SSH is a TCP protocol. So don't change this line.

- e. Upload an SSH public key by appending the following code:

```
# Specify an SSH public key
resource "opc_compute_ssh_key" "default" {
  name = "occKey"
  key  = "ssh-rsa AAAAB3NzaC1yc2E..."
  lifecycle {
    prevent_destroy = true
  }
}
```

In this code:

- Don't change the `resource` line.
- `name`: Replace with a name of your choice, or leave the example as is.

- `key`: Replace with the value of your SSH public key. Copy and paste the value exactly as in the public key file. Don't introduce any extra characters or lines.
- `lifecycle.prevent_destroy=true` ensures that the resource is retained even when you delete the VM.

f. Create a virtual NIC set by appending the following code:

```
# Create a virtual NIC set
resource "opc_compute_vnic_set" "default" {
  name          = "occVNICset"
  applied_acls = ["${opc_compute_acl.default.name}"]
}
```

In this code:

- Don't change the `resource` line.
- `name`: Replace with a name of your choice, or leave the example as is.
- `applied_acls` contains a reference to the ACL that you defined earlier.

g. Define a security rule to permit SSH access to the VM by appending the following code:

```
resource "opc_compute_security_rule" "default" {
  name              = "occSecurityRule"
  flow_direction   = "ingress"
  acl               = "${opc_compute_acl.default.name}"
  security_protocols = ["${
opc_compute_security_protocol.default.name}"]
  dst_vnic_set     = "${opc_compute_vnic_set.default.name}"
}
```

In this code:

- Don't change the `resource` line.
- `name`: Replace with a name of your choice, or leave the example as is.
- `flow_direction=ingress` means that this rule permits inbound traffic to the VM. Don't change this line.
- `acl` contains a reference to the ACL that you defined earlier. Don't change this line.
- `security_protocols` contains a reference to the SSH protocol that you defined earlier. Don't change this line.
- `dst_vnic_set` contains a reference to the vNICset that you defined earlier. Don't change this line.

h. Create a persistent boot volume using the Oracle Linux 7.2 image, by appending the following code:

```
# Create a persistent boot volume
resource "opc_compute_storage_volume" "boot" {
  size = "20"
  name = "occBootVolume"
```

```
bootable = true
image_list = "/oracle/public/OL_7.2_UEKR4_x86_64"
image_list_entry = 1
lifecycle {
  prevent_destroy = true
}
}
```

In this code:

- Don't change the `resource` line.
 - `size`: Leave it at 20 GB or enter a larger size.
 - `name`: Replace with a name of your choice, or leave the example as is.
 - `bootable=true` means that this a bootable volume. Don't change this line.
 - `image_list`: Replace with the full name of the image that you want to use, or leave the example as is.
 - `image_list_entry=1` means that the first image in the image list must be used. Don't change this line.
 - `lifecycle.prevent_destroy=true` ensures that the resource is retained even when you delete the VM.
- i. Create a volume for data and applications that you may want to store, by appending the following code:

```
# Create a data volume
resource "opc_compute_storage_volume" "data" {
  name = "occDataVolume"
  size = 10
  lifecycle {
    prevent_destroy = true
  }
}
```

In this code:

- Don't change the `resource` line.
 - `name`: Replace with a name of your choice, or leave the example as is.
 - `size`: Replace with a size of your choice, in GB.
 - `lifecycle.prevent_destroy=true` ensures that the resource is retained even when you delete the VM.
- j. Create a VM by appending the following code:

```
# Create a VM
resource "opc_compute_instance" "default" {
  name      = "occVM"
  shape     = "oc3"
  ssh_keys = ["${opc_compute_ssh_key.default.name}"]
  hostname = "ocvm"

  storage {
```



```
        volume = "${opc_compute_storage_volume.boot.name}"
        index   = 1
    }

    boot_order = [1]

    storage {
        volume = "${opc_compute_storage_volume.data.name}"
        index  = 2
    }

    networking_info {
        index      = 0
        ip_network = "${opc_compute_ip_network.default.name}"
        nat        = ["${
opc_compute_ip_address_reservation.default.name}"]
        vnic_sets = ["${opc_compute_vnic_set.default.name}"]
    }
}
```

In this code:

- Don't change the `resource` line.
 - `name`: Replace with a name of your choice, or leave the example as is.
 - `shape`: Replace with a shape of your choice, or leave the example as is.
 - `ssh_keys` contains a reference to the SSH public key that you specified earlier. Don't change this line.
 - `storage.volume`: There are two of these fields referring to the boot and data volumes that you defined earlier. Don't change these lines.
 - `storage.index` indicates the disk number at which volume must be attached to the VM. Don't change these lines.
 - `boot_order=1` means that the volume attached at index #1 must be used to boot the VM. Don't change this line.
 - `networking_info.index=0` means that this network definition is for `eth0`. Don't change this line.
 - `networking_info.ip_network` contains a reference to the IP network that you defined earlier. Don't change this line.
 - `networking_info.nat` contains a reference to the IP reservation that you defined earlier. Don't change this line.
 - `networking_info.vnic_sets` contains a reference to the vNICset that you defined earlier. Don't change this line.
6. After adding all the required code, save the file.
 7. Initialize the directory containing the configuration.

```
terraform init
```

This command downloads the `opc` provider and sets up the current directory for use by Terraform.

8. Verify that the syntax of the configuration has no errors.

```
terraform validate
```

If any syntactical errors exist, the output lists the errors.

9. If errors exist, then reopen the configuration, fix the errors, save the file, and run `terraform validate` again.

When no error exists, the command doesn't display any output.

 **Tip:**

To debug problems from this point onward, you can enable logging.

- a. Configure the log level by setting the `TF_LOG` environment variable to `TRACE`, `DEBUG`, `INFO`, `WARN` or `ERROR`. The `TRACE` level is the most verbose.
- b. Set the log-file path by using the `TF_LOG_PATH` environment variable.

10. Review the resources that you have defined.

```
terraform plan
```

Terraform displays all the actions that will be performed when you apply this configuration. It lists the resources that will be created and deleted and the attributes of each resource. Here's an example of the output of the command.

 **Note:**

In this example, some parts are truncated for brevity, and only the attributes that you defined explicitly are shown. When you run `terraform plan`, you'll see several more attributes with the value `<computed>`. The values for those fields will be filled when the resources are created.

Terraform will perform the following actions:

```
+ opc_compute_acl.default
  enabled: "true"
  name: "occACL"

+ opc_compute_instance.default
  boot_order.0: "1"
  name: "occVM"
  networking_info.2552438773.index: "0"
  networking_info.2552438773.ip_network: "occIPnetwork"
  networking_info.2552438773.nat.0: "occIPreservation"
  networking_info.2552438773.vnic_sets.0: "occVNICset"
  shape: "oc3"
  ssh_keys.0: "occKey"
  storage.1528687378.index: "2"
  storage.1528687378.volume: "occDataVolume"
```

```
storage.3242904380.index: "1"
storage.3242904380.volume: "occBootVolume"

+ opc_compute_ip_address_reservation.default
  ip_address_pool: "public-ippool"
  name: "occIPreservation"

+ opc_compute_ip_network.default
  ip_address_prefix: "192.168.100.0/24"
  name: "occIPnetwork"

+ opc_compute_security_protocol.default
  dst_ports.0: "22"
  ip_protocol: "tcp"
  name: "occSSHprotocol"

+ opc_compute_security_rule.default
  acl: "occACL"
  dst_vnic_set: "occVNICset"
  flow_direction: "ingress"
  name: "occSecurityRule"
  security_protocols.0: "occSSHprotocol"

+ opc_compute_ssh_key.default
  key: "ssh-rsa"
  AAAAB3NzaC1yc2EAAAAB...
  name: "occKey"

+ opc_compute_storage_volume.boot
  bootable: "true"
  image_list: "/oracle/"
  public/OL_7.2_UEKR4_x86_64"
  image_list_entry: "1"
  name: "occBootVolume"
  size: "20"

+ opc_compute_storage_volume.data
  name: "occDataVolume"
  size: "10"

+ opc_compute_vnic_set.default
  applied_acls.0: "occACL"
  name: "occVNICset"
```

At the end, Terraform summarizes the number of resources that will be added, destroyed, and changed when you apply the configuration.

Plan: 10 to add, 0 to change, 0 to destroy.

11. If you want to change anything, edit the configuration, validate it, and review the revised plan.
12. After finalizing the configuration, create the resources defined in it.

```
terraform apply
```

13. At the "Do you want to perform these actions" prompt, enter **yes**.

Terraform displays the status of the operation, as shown in the following example. For each resource, Terraform shows the status and the time taken for the operation.

 **Note:**

In this example, some parts are truncated for brevity.

```
opc_compute_security_protocol.default: Creating...
opc_compute_ssh_key.default: Creating...
opc_compute_storage_volume.data: Creating...
opc_compute_storage_volume.boot: Creating...
opc_compute_ip_network.default: Creating...
opc_compute_acl.default: Creating...
opc_compute_ip_address_reservation.default: Creating...
opc_compute_ip_network.default: Creation complete after 1s (ID:
occIPnetwork)
opc_compute_security_protocol.default: Creation complete after 1s (ID:
occSSHprotocol)
opc_compute_ip_address_reservation.default: Creation complete after 1s
(ID: occIPreservation)
opc_compute_acl.default: Creation complete after 1s (ID: occACL)
opc_compute_vnic_set.default: Creating...
opc_compute_vnic_set.default: Creation complete after 0s (ID: occVNICset)
opc_compute_security_rule.default: Creating...
opc_compute_security_rule.default: Creation complete after 1s (ID:
occSecurityRule)
opc_compute_ssh_key.default: Creation complete after 2s (ID: occKey)
opc_compute_storage_volume.data: Creation complete after 12s (ID:
occDataVolume)
opc_compute_storage_volume.boot: Creation complete after 12s (ID:
occBootVolume)
opc_compute_instance.default: Creating...
opc_compute_instance.default: Creation complete after 43s (ID: 9a3fee81-
b742-48f3-be2d-b83b842e3b40)
```

14. Wait for the following message:

```
Apply complete! Resources: 10 added, 0 changed, 0 destroyed.
```

15. Find out the public IP address of the VM.

```
terraform state show opc_compute_ip_address_reservation.default
```

Here's an example of the output of this command:

```
id           = occIPreservation
description  =
ip_address   = 198.51.100.1
ip_address_pool = public-ippool
name         = occIPreservation
tags.#      = 0
uri         = https://203.0.100.1/network/v1/ipreservation/
Compute-500099999/jack.smith@example.com/occIPreservation
```

In the output, look for the `ip_address` field. You can use this address to connect to the VM using `ssh`.

Add, Update, and Delete Resources Using Terraform

You can manage your IaaS and PaaS resources on Oracle Cloud at Customer by using the Terraform configuration that you used originally to create the resources.

Add Resources

Define the required resources in the configuration, and run `terraform apply`.

Update Resources

Edit the attributes of the resources in the configuration, and run `terraform apply`.

Delete Resources

- To delete a specific resource, run the following command:

```
terraform destroy -target=resource_type.resource_name
```

For example, to delete just the VM in the configuration that you applied earlier, run this command:

```
terraform destroy -target=opc_compute_instance.default
```

At the "Do you really want to destroy" prompt, enter **yes**.

Terraform displays the status of the operation, as shown in the following example. For each resource, Terraform shows the status and the time taken for the operation.

 **Note:**

In this example, some parts are truncated for brevity.

```
opc_compute_instance.default: Destroying... (ID: 9a3fee81-b742-48f3-
be2d-b83b842e3b40)
...
```

```
...
opc_compute_instance.default: Destruction complete after 41s
```

Wait for the following message:

```
Destroy complete! Resources: 1 destroyed
```

- To delete all the resources, run `terraform destroy`.
- To delete specific resources permanently, remove the resources from the configuration, and then run `terraform apply`.

Re-create Resources

To re-create any resources that you deleted previously but didn't remove from the configuration, run `terraform apply`.

Learn More

- For information about the resources and configuration options that Terraform supports for the `opc` provider, see the Terraform documentation at <https://www.terraform.io/docs/providers/opc/index.html>.
- For help with the Terraform CLI commands, see <https://www.terraform.io/docs/commands/index.html>.

Create a Multi-Tier Topology with IP Networks Using Terraform

Terraform is a third-party tool that you can use to create and manage your IaaS and PaaS resources on Oracle Cloud at Customer. This guide shows you how to use Terraform to launch and manage a multi-tier topology of Compute Classic instances attached to IP networks.

Topics

- [Scenario Overview](#)
- [Prerequisites](#)
- [Create the Required Resources Using Terraform](#)
- [\(Optional\) Verify Network Access to the VMs](#)

Scenario Overview

The application and the database that the application uses are hosted on instances attached to separate IP networks. Users outside Oracle Cloud have HTTPS access to the application instances. The topology also includes an admin instance that users outside the cloud can connect to using SSH. The admin instance can communicate with all the other instances in the topology.

 **Note:**

The focus of this guide is the network configuration for instances attached to IP networks in a sample topology. The framework and the flow of the steps can be applied to other similar or more complex topologies. The steps for provisioning or configuring other resources (like storage) are not covered in this guide.

Compute Topology

The topology that you are going to build using the steps in this tutorial contains the following Compute Classic instances:

- Two instances – `appVM1` and `appVM2` – for hosting a business application, attached to an IP network, each with a fixed public IP address.
- Two instances – `dbVM1` and `dbVM2` – for hosting the database for the application. These instances are attached to a second IP network.
- An admin instance – `adminVM` – that's attached to a third IP network and has a fixed public IP address.

 **Note:**

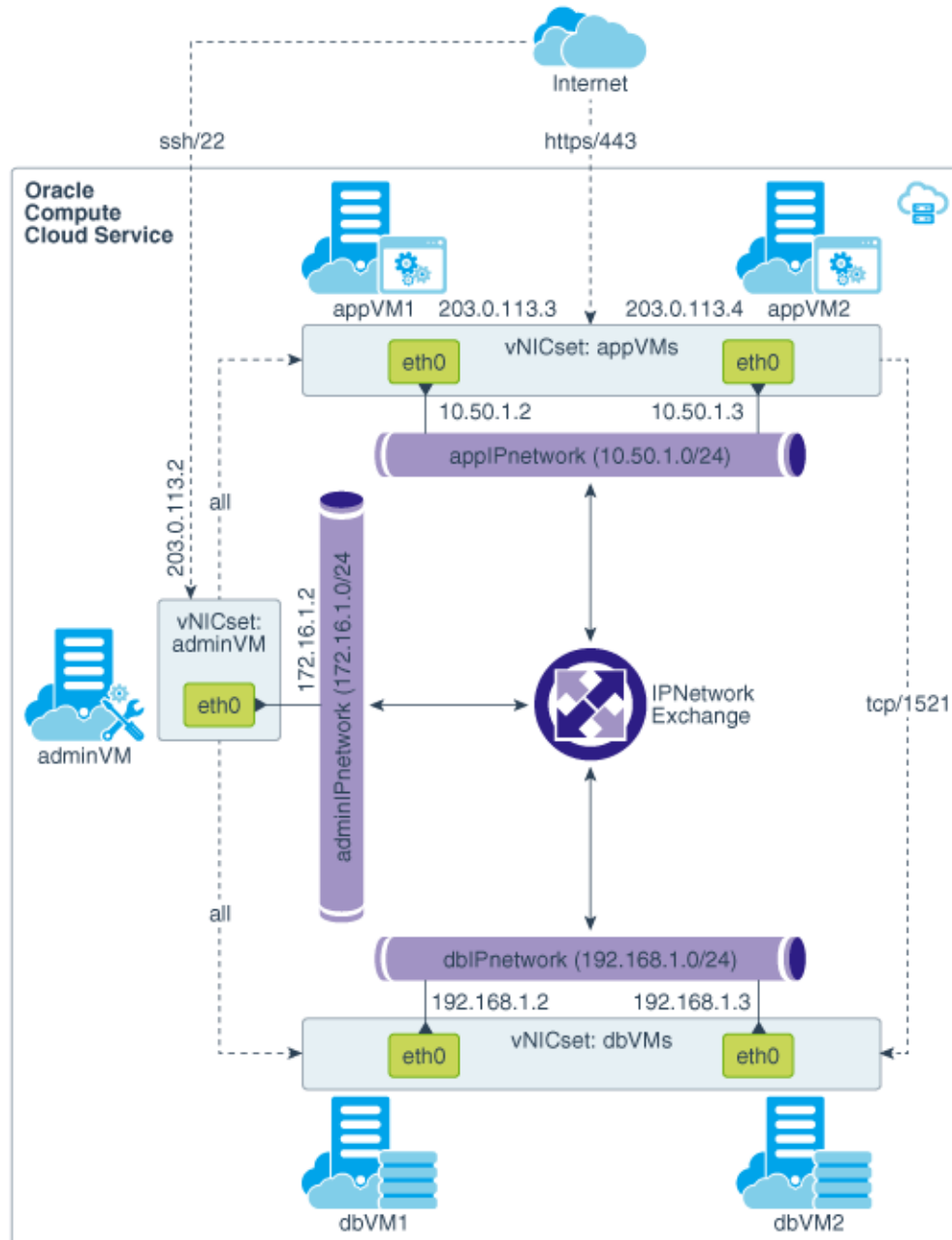
You won't actually install any application or database. Instead, you'll simulate listeners on the required application and database ports using the `nc` utility. The goal of this section is to demonstrate the steps to configure the networking that's necessary for the traffic flow requirements described next.

Traffic Flow Requirements

Only the following traffic flows must be permitted in the topology that you'll build:

- HTTPS requests from anywhere to the application instances
- SSH connections from anywhere to the admin instance
- All traffic from the admin instance to the application instances
- All traffic from the admin instance to the database instances
- TCP traffic from two application instances to port 1521 of the database instances

Topology Architecture Diagram



Network Resources Required for this Topology

- **Public IP address reservations** for the application instances and for the admin instance
- Three **IP networks**, one each for the application instances, the database instances, and the admin instance
- An **IP network exchange** to connect the IP networks in the topology
- **Security protocols** for SSH, HTTPS, and TCP/1521 traffic
- **ACLs** that will contain the required security rules
- **vNICsets** for the application instances, database instances, and the admin instance

- **Security rules** to allow SSH connections to the admin instance, HTTPS traffic to the application instances, and TCP/1521 traffic to the database instances

Prerequisites

1. If you are new to Terraform, learn the basics.
At a minimum, read the brief introduction here: <https://www.terraform.io/intro/index.html>.
2. Download and install Terraform on your local computer.
Binary packages are available for several operating systems and processor architectures. For the instructions to download and install Terraform, go to <https://www.terraform.io/intro/getting-started/install.html>.
3. Generate an SSH key pair. See [Generate an SSH Key Pair](#).
4. Gather the required Oracle Cloud account information:
 - Your Oracle Cloud user name and password.
 - The service instance ID.
 - a. Sign in to Oracle Cloud My Services.
 - b. Locate the Compute Classic tile and click **Compute Classic**.
 - c. Locate the **Service Instance ID** field, and note its value (example: 500099999).
 - The REST endpoint URL for Compute Classic.
 - a. Sign in to Oracle Cloud My Services, using the My Services URL from the welcome email.
 - b. Click ☰ near the upper left corner of the page.
 - c. In the menu that appears, expand **Services**, and click **Compute Classic**. The Instances page of the Compute Classic web console is displayed.
 - d. Click **Site** near the top of the page, and select the site for which you want to find out the REST endpoint URL.
 - e. In the **Site Selector** dialog box, note the URL in the REST Endpoint field.

Create the Required Resources Using Terraform

Define all the resources required for the multi-tier topology in a Terraform configuration and then apply the configuration.



Note:

The procedure described here shows how to define resources in a simple Terraform configuration. It does not use advanced Terraform features, such as variables and modules.

1. On the computer where you installed Terraform, create a new directory.
2. In the new directory, create an empty text file, *name-of-your-choice.tf*.

This is a Terraform configuration. In this file, you define the following:

- The parameters that Terraform must use to connect to your Oracle Cloud at Customer machine
- The resources to be provisioned

 **Important:**

The `.tf` extension is mandatory. When Terraform performs any operation, it looks for a file with the `.tf` extension in the current directory.

3. Open the text file in an editor of your choice.
4. Add the following code to define the parameters that Terraform needs to connect to your account:

```
provider "opc" {  
  user          = "jack.smith@example.com"  
  password      = "mypassword"  
  identity_domain = "500099999"  
  endpoint      = "https://compute.site99.ocm.rack100.example.com"  
}
```

In this code:

- Don't change the `provider` line.
 - `user` and `password`: Replace with your Oracle Cloud credentials.
 - `identity_domain`: Replace with the service instance ID that you identified earlier.
 - `endpoint`: Replace with the REST endpoint URL of Compute Classic.
5. Add code for each resource that you want to create using Terraform.

 **Note:**

When copying and editing the code, follow the instructions carefully.

- a. Add code for the ACLs:

```
# Create the ACLs  
  
# For the admin VM  
resource "opc_compute_acl" "adminVM" {  
  name = "adminVM"  
}  
  
# For the application VMs  
resource "opc_compute_acl" "appVMs" {  
  name = "appVMs"  
}  
  
# For the database VMS  
resource "opc_compute_acl" "dbVMs" {
```

```

    name = "dbVMs"
  }

```

In this code:

- Don't change the `resource` lines.
- `name`: Replace with names of your choice, or leave the examples as is.

b. Add code for an IP exchange:

```

# Create an IP exchange
resource "opc_compute_ip_network_exchange" "occIPX" {
  name = "occIPX"
}

```

In this code:

- Don't change the `resource` line.
- `name`: Replace with a name of your choice, or leave the example as is.

c. Add code for the IP networks:

```

# Create the IP networks

# For the admin VM
resource "opc_compute_ip_network" "adminIPnetwork" {
  name                = "adminIPnetwork"
  ip_network_exchange = "$
{opc_compute_ip_network_exchange.occIPX.name}"
  ip_address_prefix  = "172.16.1.0/24"
}

# For the application VMs
resource "opc_compute_ip_network" "appIPnetwork" {
  name                = "appIPnetwork"
  ip_network_exchange = "$
{opc_compute_ip_network_exchange.occIPX.name}"
  ip_address_prefix  = "10.50.1.0/24"
}

# For the database VMs
resource "opc_compute_ip_network" "dbIPnetwork" {
  name                = "dbIPnetwork"
  ip_network_exchange = "$
{opc_compute_ip_network_exchange.occIPX.name}"
  ip_address_prefix  = "192.168.1.0/24"
}

```

In this code:

- Don't change the `resource` lines.
- `name`: Replace with names of your choice, or leave the examples as is.
- `ip_network_exchange` is a reference to the IP network exchange that you defined earlier. Don't change these lines.

- `ip_address_prefix`: Replace with address ranges of your choice in CIDR format, or leave the examples as is.
- d. Add code to reserve public IP addresses for the VMs:

```
# Reserve public IP addresses

# For the admin VM
resource "opc_compute_ip_address_reservation" "ipResForAdminVM" {
  name           = "ipResForAdminVM"
  ip_address_pool = "public-ippool"
  lifecycle {
    prevent_destroy = true
  }
}

# For application VM 1
resource "opc_compute_ip_address_reservation" "ipResForAppVM1" {
  name           = "ipResForAppVM1"
  ip_address_pool = "public-ippool"
  lifecycle {
    prevent_destroy = true
  }
}

# For application VM 2
resource "opc_compute_ip_address_reservation" "ipResForAppVM2" {
  name           = "ipResForAppVM2"
  ip_address_pool = "public-ippool"
  lifecycle {
    prevent_destroy = true
  }
}
```

In this code:

- Don't change the `resource` lines.
 - `name`: Replace with names of your choice, or leave the examples as is.
 - `ip_address_pool`: Don't change these lines.
 - `lifecycle.prevent_destroy=true` reduces the chance of accidentally deleting the resource. This setting is useful for resources that you want to retain for future use even after you delete the VM.
- e. Add code for the required security protocols:

```
# Create security protocols

# For HTTPS requests to the application VMs
resource "opc_compute_security_protocol" "https" {
  name           = "https"
  dst_ports     = ["443"]
  ip_protocol   = "tcp"
}

# For SSH connections
resource "opc_compute_security_protocol" "ssh" {
```

```

    name          = "ssh"
    dst_ports     = ["22"]
    ip_protocol   = "tcp"
  }

  # For TCP traffic from the application VMs to the database VMs
  resource "opc_compute_security_protocol" "tcp1521" {
    name          = "tcp1521"
    dst_ports     = ["1521"]
    ip_protocol   = "tcp"
  }

```

In this code:

- Don't change the `resource` lines.
- `name`: Replace with names of your choice, or leave the examples as is.
- `dst_ports`: 443, 22, and 1521 are the ports we need to open. Don't change these lines.
- `ip_protocol`: TCP is the protocol for all the ports that we need to open. Don't change these lines.

f. Add code to upload an SSH public key:

```

# Specify an SSH public key
resource "opc_compute_ssh_key" "adminSSHkey" {
  name = "ocKey"
  key  = "ssh-rsa AAAAB3NzaClyc2E..."
  lifecycle {
    prevent_destroy = true
  }
}

```

In this code:

- Don't change the `resource` line.
- `name`: Replace with a name of your choice, or leave the example as is.
- `key`: Replace with the value of your SSH public key. Copy and paste the value exactly as in the public key file. Don't introduce any extra characters or lines.
- `lifecycle.prevent_destroy=true` ensures that the resource is retained even when you delete the VM.

g. Add code for the virtual NIC sets:

```

# Create virtual NIC sets

# For the admin VM
resource "opc_compute_vnic_set" "adminVM" {
  name          = "adminVM"
  applied_acls = ["${opc_compute_acl.adminVM.name}"]
}

```

```
# For the application VMs
resource "opc_compute_vnic_set" "appVMs" {
  name      = "appVMs"
  applied_acls = ["${opc_compute_acl.appVMs.name}"]
}

# For the database VMs
resource "opc_compute_vnic_set" "dbVMs" {
  name      = "dbVMs"
  applied_acls = ["${opc_compute_acl.dbVMs.name}"]
}
```

In this code:

- Don't change the `resource` lines.
- `name`: Replace with names of your choice, or leave the examples as is.
- `applied_acls` contain references to the ACLs that you defined earlier. Don't change these lines.

h. Add code for the following security rules:

Purpose	Suggested Name	Type	ACL	Source and Destination	Protocol
SSH requests from any source to the admin VM	internet-to-adminVM	Ingress	adminVM	Source: Any Destination: adminVM vNICset	ssh
All traffic from the admin VM to any destination	adminVM-to-any	Egress	adminVM	Source: adminVM vNICset Destination: Any	Any
All traffic from the admin VM to the application VMs	adminVM-to-appVMs	Ingress	appVMs	Source: adminVM vNICset Destination: appVMs vNICset	Any
HTTPS traffic from any source to port 443 of the application VMs	internet-to-appVMs	Ingress	appVMs	Source: Any Destination: appVMs vNICset	https
TCP traffic from the application VMs to port 1521 of the DB VMs	appVMs-to-dbVMs-egress	Egress	appVMs	Source: appVMs vNICset Destination: dbVMs vNICset	tcp1521
TCP traffic from the application VMs to port 1521 of the DB VMs	appVMs-to-dbVMs-ingress	Ingress	dbVMs	Source: appVMs vNICset Destination: dbVMs vNICset	tcp1521
All traffic from the admin VM to the DB VMs	adminVM-to-dbVMs	Ingress	dbVMs	Source: adminVM vNICset Destination: dbVMs vNICset	Any

```
# Create security rules
```

```
# For SSH requests from any source to the admin VM
resource "opc_compute_security_rule" "internet-to-adminVM" {
```

```

    name                = "internet-to-adminVM"
    flow_direction      = "ingress"
    acl                 = "${opc_compute_acl.adminVM.name}"
    security_protocols = ["$
{opc_compute_security_protocol.ssh.name}"]
    dst_vnic_set       = "${opc_compute_vnic_set.adminVM.name}"
  }

# For all traffic from the admin VM to any destination
resource "opc_compute_security_rule" "adminVM-to-any" {
  name                = "adminVM-to-any"
  flow_direction      = "egress"
  acl                 = "${opc_compute_acl.adminVM.name}"
  src_vnic_set       = "${opc_compute_vnic_set.adminVM.name}"
}

# For all traffic from the admin VM to the application VMs
resource "opc_compute_security_rule" "adminVM-to-appVMs" {
  name                = "adminVM-to-appVMs"
  flow_direction      = "ingress"
  acl                 = "${opc_compute_acl.appVMs.name}"
  src_vnic_set       = "${opc_compute_vnic_set.adminVM.name}"
  dst_vnic_set       = "${opc_compute_vnic_set.appVMs.name}"
}

# For HTTPS traffic from any source to port 443 of the
application VMs
resource "opc_compute_security_rule" "internet-to-appVMs" {
  name                = "internet-to-appVMs"
  flow_direction      = "ingress"
  acl                 = "${opc_compute_acl.appVMs.name}"
  security_protocols = ["$
{opc_compute_security_protocol.https.name}"]
  dst_vnic_set       = "${opc_compute_vnic_set.appVMs.name}"
}

# For TCP traffic from the application VMs to port 1521 of the
DB VMs
resource "opc_compute_security_rule" "appVMs-to-dbVMs-egress" {
  name                = "appVMs-to-dbVMs-egress"
  flow_direction      = "egress"
  acl                 = "${opc_compute_acl.appVMs.name}"
  security_protocols = ["$
{opc_compute_security_protocol.tcp1521.name}"]
  src_vnic_set       = "${opc_compute_vnic_set.appVMs.name}"
  dst_vnic_set       = "${opc_compute_vnic_set.dbVMs.name}"
}

# For TCP traffic from the application VMs to port 1521 of the
DB VMs
resource "opc_compute_security_rule" "appVMs-to-dbVMs-ingress" {
  name                = "appVMs-to-dbVMs-ingress"
  flow_direction      = "ingress"
  acl                 = "${opc_compute_acl.dbVMs.name}"
  security_protocols = ["$

```

```

{opc_compute_security_protocol.tcp1521.name}]
  src_vnic_set      = "${opc_compute_vnic_set.appVMs.name}"
  dst_vnic_set      = "${opc_compute_vnic_set.dbVMs.name}"
}

# For all traffic from the admin VM to the DB VMs
resource "opc_compute_security_rule" "adminVM-to-dbVMs" {
  name                = "adminVM-to-dbVMs"
  flow_direction      = "ingress"
  acl                 = "${opc_compute_acl.dbVMs.name}"
  src_vnic_set        = "${opc_compute_vnic_set.adminVM.name}"
  dst_vnic_set        = "${opc_compute_vnic_set.dbVMs.name}"
}

```

In this code:

- Don't change the `resource` lines.
- `name`: Replace with names of your choice, or leave the examples as is.
- `flow_direction` is the direction (to or from the VMs) in which the rules permit traffic. Don't change these line.
- `acl` is a reference to one of the ACLs that you defined earlier. Don't change these lines.
- `security_protocols` are references to the protocols that you defined earlier. Don't change these lines.
- `src_vnic_set` and `dst_vnic_set` are references to the appropriate vNICsets that you defined earlier. Don't change these lines.

i. Add code to create persistent boot volumes for the VMs:

```

# Create persistent boot volumes

# For the admin VM
resource "opc_compute_storage_volume" "adminVMbootVolume" {
  size = "20"
  name = "adminVMbootVolume"
  bootable = true
  image_list = "/oracle/public/OL_7.2_UEKR4_x86_64"
  image_list_entry = 1
  lifecycle {
    prevent_destroy = true
  }
}

# For application VM 1
resource "opc_compute_storage_volume" "appVM1bootVolume" {
  size = "20"
  name = "appVM1bootVolume"
  bootable = true
  image_list = "/oracle/public/OL_7.2_UEKR4_x86_64"
  image_list_entry = 1
  lifecycle {
    prevent_destroy = true
  }
}

```



```
}

# For application VM 2
resource "opc_compute_storage_volume" "appVM2bootVolume" {
  size = "20"
  name = "appVM2bootVolume"
  bootable = true
  image_list = "/oracle/public/OL_7.2_UEKR4_x86_64"
  image_list_entry = 1
  lifecycle {
    prevent_destroy = true
  }
}

# For database VM 1
resource "opc_compute_storage_volume" "dbVM1bootVolume" {
  size = "20"
  name = "dbVM1bootVolume"
  bootable = true
  image_list = "/oracle/public/OL_7.2_UEKR4_x86_64"
  image_list_entry = 1
  lifecycle {
    prevent_destroy = true
  }
}

# For database VM 2
resource "opc_compute_storage_volume" "dbVM2bootVolume" {
  size = "20"
  name = "dbVM2bootVolume"
  bootable = true
  image_list = "/oracle/public/OL_7.2_UEKR4_x86_64"
  image_list_entry = 1
  lifecycle {
    prevent_destroy = true
  }
}
```

In this code:

- Don't change the `resource` lines.
- `size`: Leave the sizes at 20 GB or enter a larger size.
- `name`: Replace with names of your choice, or leave the examples as is.
- `bootable=true` indicates a bootable volume. Don't change these lines.
- `image_list`: Replace with the full name of the images that you want to use, or leave the examples as is.
- `image_list_entry=1` means that the first image in the image list must be used. Don't change these lines.
- `lifecycle.prevent_destroy=true` ensures that the resource is retained even when you delete the VM.

j. Add code for volumes for the data and applications that you may want to store:

```
# Create data volumes

# For the admin VM
resource "opc_compute_storage_volume" "adminVMdataVolume" {
  name = "adminVMdataVolume"
  size = 10
  lifecycle {
    prevent_destroy = true
  }
}

# For application VM 1
resource "opc_compute_storage_volume" "appVM1dataVolume" {
  name = "appVM1dataVolume"
  size = 10
  lifecycle {
    prevent_destroy = true
  }
}

# For application VM 2
resource "opc_compute_storage_volume" "appVM2dataVolume" {
  name = "appVM2dataVolume"
  size = 10
  lifecycle {
    prevent_destroy = true
  }
}

# For database VM 1
resource "opc_compute_storage_volume" "dbVM1dataVolume" {
  name = "dbVM1dataVolume"
  size = 10
  lifecycle {
    prevent_destroy = true
  }
}

# For database VM 2
resource "opc_compute_storage_volume" "dbVM2dataVolume" {
  name = "dbVM2dataVolume"
  size = 10
  lifecycle {
    prevent_destroy = true
  }
}
```

In this code:

- Don't change the `resource` lines.
- `name`: Replace with names of your choice, or leave the examples as is.
- `size`: Replace with sizes of your choice, in GB.

- `lifecycle.prevent_destroy=true` ensures that the resource is retained even when you delete the VM.

k. Add code for the admin VM:

```
# Create the admin VM
resource "opc_compute_instance" "adminVM" {
  name      = "adminVM"
  shape     = "oc3"
  ssh_keys  = ["${opc_compute_ssh_key.adminSSHkey.name}"]
  hostname  = "adminvm"

  storage {
    volume = "$
{opc_compute_storage_volume.adminVMbootVolume.name}"
    index  = 1
  }

  boot_order = [1]

  storage {
    volume = "$
{opc_compute_storage_volume.adminVMdataVolume.name}"
    index  = 2
  }

  networking_info {
    index      = 0
    ip_network = "${opc_compute_ip_network.adminIPnetwork.name}"
    nat        = ["$
{opc_compute_ip_address_reservation.ipResForAdminVM.name}"]
    vnic_sets  = ["${opc_compute_vnic_set.adminVM.name}"]
  }
}
```

In this code:

- Don't change the `resource` line.
- `name`: Replace with a name of your choice, or leave the example as is.
- `shape`: Replace with a shape of your choice, or leave the example as is.
- `ssh_keys` contains a reference to the SSH public key that you specified earlier. Don't change this line.
- `hostname`: Replace with a host name of your choice, or leave the example as is.
- `storage.volume`: There are two of these fields referring to the boot and data volumes that you defined earlier. Don't change these lines.
- `storage.index` indicates the disk number at which volume must be attached to the VM. Don't change these lines.
- `boot_order=1` means that the volume attached at index #1 must be used to boot the VM. Don't change this line.

- `networking_info.index=0` means that this network definition is for `eth0`. Don't change this line.
- `networking_info.ip_network` contains a reference to the IP network that you defined earlier. Don't change this line.
- `networking_info.nat` contains a reference to the IP reservation that you defined earlier. Don't change this line.
- `networking_info.vnic_sets` contains a reference to the vNICset that you defined earlier. Don't change this line.

I. Add code for the application VMs:

```
# Create application VM 1
resource "opc_compute_instance" "appVM1" {
  name      = "appVM1"
  shape     = "oc3"
  ssh_keys = ["${opc_compute_ssh_key.adminSSHkey.name}"]
  hostname  = "appvm1"

  storage {
    volume = "${opc_compute_storage_volume.appVM1bootVolume.name}"
    index  = 1
  }

  boot_order = [1]

  storage {
    volume = "${opc_compute_storage_volume.appVM1dataVolume.name}"
    index  = 2
  }

  networking_info {
    index      = 0
    ip_network = "${opc_compute_ip_network.appIPnetwork.name}"
    nat        = ["${
opc_compute_ip_address_reservation.ipResForAppVM1.name}"]
    vnic_sets = ["${opc_compute_vnic_set.appVMs.name}"]
  }
}

# Create application VM 2
resource "opc_compute_instance" "appVM2" {
  name      = "appVM2"
  shape     = "oc3"
  ssh_keys = ["${opc_compute_ssh_key.adminSSHkey.name}"]
  hostname  = "appvm2"

  storage {
    volume = "${opc_compute_storage_volume.appVM2bootVolume.name}"
    index  = 1
  }

  boot_order = [1]

  storage {
```

```

        volume = "$
{opc_compute_storage_volume.appVM2dataVolume.name}"
        index = 2
    }

    networking_info {
        index      = 0
        ip_network = "${opc_compute_ip_network.appIPnetwork.name}"
        nat        = ["$
{opc_compute_ip_address_reservation.ipResForAppVM2.name}"]
        vnic_sets = ["${opc_compute_vnic_set.appVMs.name}"]
    }
}

```

In this code:

- Don't change the `resource` lines.
- `name`: Replace with names of your choice, or leave the examples as is.
- `shape`: Replace with shapes of your choice, or leave the examples as is.
- `ssh_keys` contain references to the SSH public key that you specified earlier. Don't change these lines.
- `hostname`: Replace with host names of your choice, or leave the examples as is.
- `storage.volume`: These fields refer to the boot and data volumes that you defined earlier. Don't change these lines.
- `storage.index` indicate the disk number at which the volumes must be attached to the VMs. Don't change these lines.
- `boot_order=1` means that the volumes attached at index #1 must be used to boot the VMs. Don't change these lines.
- `networking_info.index=0` means that the network definitions are for `eth0`. Don't change these lines.
- `networking_info.ip_network` is a reference to the IP network that you defined earlier. Don't change these lines.
- `networking_info.nat` is a reference to the IP reservation that you defined earlier for each VM. Don't change these lines.
- `networking_info.vnic_sets` are references to the vNICsets that you defined earlier. Don't change these lines.

m. Add code for the database VMs:

```

# Create database VM 1
resource "opc_compute_instance" "dbVM1" {
    name      = "dbVM1"
    shape     = "oc3"
    ssh_keys  = ["${opc_compute_ssh_key.adminSSHkey.name}"]
    hostname  = "dbvm1"

    storage {
        volume = "${opc_compute_storage_volume.dbVM1bootVolume.name}"
    }
}

```

```

        index = 1
    }

    boot_order = [1]

    storage {
        volume = "${opc_compute_storage_volume.dbVM1dataVolume.name}"
        index = 2
    }

    networking_info {
        index = 0
        ip_network = "${opc_compute_ip_network.dbIPnetwork.name}"
        vnic_sets = ["${opc_compute_vnic_set.dbVMs.name}"]
    }
}

# Create database VM 2
resource "opc_compute_instance" "dbVM2" {
    name          = "dbVM2"
    shape         = "oc3"
    ssh_keys     = ["${opc_compute_ssh_key.adminSSHkey.name}"]
    hostname     = "dbvm2"

    storage {
        volume = "${opc_compute_storage_volume.dbVM2bootVolume.name}"
        index = 1
    }

    boot_order = [1]

    storage {
        volume = "${opc_compute_storage_volume.dbVM2dataVolume.name}"
        index = 2
    }

    networking_info {
        index = 0
        ip_network = "${opc_compute_ip_network.dbIPnetwork.name}"
        vnic_sets = ["${opc_compute_vnic_set.dbVMs.name}"]
    }
}

```

In this code:

- Don't change the `resource` lines.
- `name`: Replace with names of your choice, or leave the examples as is.
- `shape`: Replace with shapes of your choice, or leave the examples as is.
- `ssh_keys` contain references to the SSH public key that you specified earlier. Don't change these lines.
- `hostname`: Replace with host names of your choice, or leave the examples as is.

- `storage.volume`: These fields refer to the boot and data volumes that you defined earlier. Don't change these lines.
 - `storage.index` indicate the disk number at which the volumes must be attached to the VMs. Don't change these lines.
 - `boot_order=1` means that the volumes attached at index #1 must be used to boot the VMs. Don't change these lines.
 - `networking_info.index=0` means that the network definitions are for `eth0`. Don't change these lines.
 - `networking_info.ip_network` is a reference to the IP network that you defined earlier. Don't change these lines.
 - `networking_info.vnic_sets` are references to the vNICsets that you defined earlier. Don't change these lines.
- n. Add code for Terraform to display the public and private IP addresses of the VMs after the configuration is applied:

```
output "adminVM public IP address" {
  value = "$
{opc_compute_ip_address_reservation.ipResForAdminVM.ip_address}"
}

output "appVM1 public IP address" {
  value = "$
{opc_compute_ip_address_reservation.ipResForAppVM1.ip_address}"
}

output "appVM2 public IP address" {
  value = "$
{opc_compute_ip_address_reservation.ipResForAppVM2.ip_address}"
}

output "appVM1 private IP address" {
  value = "${opc_compute_instance.appVM1.ip_address}"
}

output "appVM2 private IP address" {
  value = "${opc_compute_instance.appVM2.ip_address}"
}

output "dbVM1 private IP address" {
  value = "${opc_compute_instance.dbVM1.ip_address}"
}

output "dbVM2 private IP address" {
  value = "${opc_compute_instance.dbVM2.ip_address}"
}
```

In this code:

- `output` is the text label to be displayed before the IP address. Don't change these lines.

- `value` is a reference to each IP address to be displayed. Don't change these lines.

6. After adding all the required code, save the file.
7. Initialize the directory containing the configuration.

```
terraform init
```

This command downloads the `opc` provider and sets up the current directory for use by Terraform.

8. Verify that the syntax of the configuration has no errors.

```
terraform validate
```

If any syntactical errors exist, the output lists the errors.

9. If errors exist, then reopen the configuration, fix the errors, save the file, and run `terraform validate` again.

When no error exists, the command doesn't display any output.

 **Tip:**

To debug problems from this point onward, you can enable logging.

- a. Configure the log level by setting the `TF_LOG` environment variable to `TRACE`, `DEBUG`, `INFO`, `WARN` or `ERROR`. The `TRACE` level is the most verbose.
- b. Set the log-file path by using the `TF_LOG_PATH` environment variable.

10. Review the resources that you have defined.

```
terraform plan
```

Terraform displays all the actions that will be performed when you apply this configuration. It lists the resources that will be created and deleted and the attributes of each resource.

At the end, Terraform summarizes the number of resources that will be added, destroyed, and changed when you apply the configuration.

```
Plan: 39 to add, 0 to change, 0 to destroy.
```

11. If you want to change anything, edit the configuration, validate it, and review the revised plan.
12. After finalizing the configuration, create the resources defined in it.

```
terraform apply
```

13. At the `Do you want to perform these actions` prompt, enter **yes**.

For each resource, Terraform shows the status of the operation and the time taken.

14. Wait for a message as shown in the following example:

```
Apply complete! Resources: 39 added, 0 changed, 0 destroyed.
```

```
Outputs:
```

```
adminVM public IP address = 203.0.113.2
```

```
appVM1 private IP address = 10.50.1.2
```

```
appVM1 public IP address = 203.0.113.3
```

```
appVM2 private IP address = 10.50.1.3
```

```
appVM2 public IP address = 203.0.113.4
```

```
dbVM1 private IP address = 192.168.1.2
```

```
dbVM2 private IP address = 192.168.1.3
```

15. Note the IP addresses. You'll need them to verify network access to the VMs.

(Optional) Verify Network Access to the VMs

Verify SSH Connections from Outside the Cloud to the Admin VM

Run the following command from your local machine:

```
[localmachine ~]$ ssh -i path-to-privateKeyFile  
opc@publicIPAddressOfAdminVM
```

You should see the following prompt:

```
opc@adminvm
```

This confirms that SSH connections can be made from outside the cloud to the admin VM.

Verify SSH Connections from the Admin VM to the Database and Application VMs

1. Copy the private SSH key file corresponding to the public key that you associated with your VMs from your local machine to the admin VM, by running the following command on your local machine:

```
[localmachine ~]$ scp -i path-to-privateKeyFile path-to-privateKeyFile opc@publicIPAddressOfAdminVM:~/.ssh/privatekey
```

2. From your local machine, connect to the admin VM using SSH:

```
[localmachine ~]$ ssh -i path-to-privateKeyFile  
opc@publicIPAddressOfAdminVM
```

3. From the admin VM, connect to each of the database and application VMs using SSH:

```
[opc@adminvm]$ ssh -i ~/.ssh/privatekey opc@privateIPAddress
```

- Depending on the VM you connect to, you should see one of the following prompts after the `ssh` connection is established.

- `opc@appvm1`
- `opc@appvm2`
- `opc@dbvm1`
- `opc@dbvm2`

Verify Connectivity from Outside the Cloud to Port 443 of the Application VMs

You can use the `nc` utility to simulate a listener on port 443 on one of the application VMs, and then run `nc` from any host outside the cloud to verify connectivity to the application VM.



Note:

The verification procedure described here is specific to VMs created using the Oracle-provided images for Oracle Linux 7.2 and 6.8.

- On your local host, download the `nc` package from http://yum.oracle.com/repo/OracleLinux/OL6/latest/x86_64/getPackage/nc-1.84-24.el6.x86_64.rpm.
- Copy `nc-1.84-24.el6.x86_64.rpm` from your local host to the admin VM.

```
[localmachine ~]$ scp -i path-to-privateKeyFile  
path_to_nc-1.84-24.el6.x86_64.rpm opc@publicIPAddressOfAdminVM:~
```

- From your local machine, connect to the admin VM using SSH:

```
[localmachine ~]$ ssh -i path-to-privateKeyFile  
opc@publicIPAddressOfAdminVM
```

- Copy `nc-1.84-24.el6.x86_64.rpm` to one of the application VMs.

```
[opc@adminvm]$ scp -i ~/.ssh/privatekey ~/nc-1.84-24.el6.x86_64.rpm  
opc@privateIPAddressOfAppVM1:~
```

- Connect to the application VM:

```
[opc@adminvm]$ ssh -i ~/.ssh/privatekey opc@privateIPAddressOfAppVM1
```

- On the application VM, install `nc`.

```
[opc@appvm1]$ sudo rpm -i nc-1.84-24.el6.x86_64.rpm
```

- Configure the application VM to listen on port 443. Note that this step is just for verifying connections to port 443. In real-life scenarios, this step would be done when you configure your application on the VM to listen on port 443.

```
[opc@appvm1]$ sudo nc -l 443
```

- From any host outside the cloud, run the following `nc` command to test whether you can connect to port 443 of the application VM:

```
[localmachine ~]$ nc -v publicIPAddressOfAppVM1 443
```

The following message is displayed:

```
Connection to publicIPAddressOfAppVM1 443 port [tcp/https]
succeeded!
```

This message confirms that the application VM accepts connection requests on port 443.

- Press `Ctrl + C` to exit the `nc` process.

Verify Connectivity from the Application VMs to Port 1521 of the Database VMs

You can use the `nc` utility to simulate a listener on port 1521 on one of the database VMs, and then run `nc` from one of the application VMs to verify connectivity from the application tier to the database tier.



Note:

The verification procedure described here is specific to VMs created using the Oracle-provided images for Oracle Linux 7.2 and 6.8.

- From your local machine, connect to the admin VM using SSH:

```
[localmachine ~]$ ssh -i path-to-privateKeyFile
opc@publicIPAddressOfAdminVM
```

- Copy `nc-1.84-24.el6.x86_64.rpm` to one of the database VMs.

```
[opc@adminvm]$ scp -i ~/.ssh/privatekey ~/nc-1.84-24.el6.x86_64.rpm
opc@privateIPAddressOfDBVM1:~
```

- Connect to the database VM:

```
[opc@adminvm]$ ssh -i ~/.ssh/privatekey opc@privateIPAddressOfDBVM1
```

- On the database VM, install `nc`.

```
[opc@dbvm1]$ sudo rpm -i nc-1.84-24.el6.x86_64.rpm
```

- Configure the VM to listen on port 1521. Note that this step is just for verifying connections to port 1521. In real-life scenarios, this step would be done when you set up your database to listen on port 1521.

```
[opc@dbvm1]$ nc -l 1521
```

6. Leave the current terminal session open. Using a new terminal session, connect to the admin VM using SSH and, from there, connect to one of the application VMs.

```
[localmachine ~]$ ssh -i path-to-privateKeyFile
opc@publicIPAddressOfAdminVM
[opc@adminvm]$ ssh -i ~/.ssh/privatekey opc@privateIPAddressOfAppVM1
```

7. From the application VM, run the following `nc` command to test whether you can connect to port 1521 of the database VM:

```
[opc@appvm1 ~]$ nc -v privateIPAddressOfDBVM1 1521
```

The following message is displayed:

```
Connection to privateIPAddressOfDBVM1 1521 port [tcp/ncube-lm] succeeded!
```

This message confirms that the database VM accepts connection requests received on port 1521 from the application VMs.

8. Press `Ctrl + C` to exit the `nc` process.

11

Compute Classic: Using the REST API

You can use the REST API to create and manage all the Compute Classic resources programmatically.

Note:

This section provides the steps to help you get started with a few basic operations. It doesn't cover all the operations that the REST API supports. For complete reference information, see *REST API for Oracle Cloud Infrastructure Compute Classic*.

Topics

- [Prepare to Use the REST API](#)
- [Get an Authentication Token](#)
- [Get the Details of a VM Using the REST API](#)
- [Add Block Storage for a VM Using the REST API](#)

Prepare to Use the REST API

Before you use the REST API, complete the following preparatory steps:

1. Install cURL.

Note:

You can access the REST API from any application or programming platform that correctly and completely understands the Hypertext Transfer Protocol (HTTP) and has Internet connectivity. cURL is a command-line tool that you can use to invoke REST API calls by sending HTTP requests. The examples in this document use the cURL command-line tool to demonstrate how to access the Compute Classic REST API.

cURL is available by default on most UNIX-like hosts.


To install cURL on Windows, complete the following steps:

- a. In your browser, navigate to <http://curl.haxx.se/download.html>.
- b. Locate the SSL-enabled version of the cURL software that corresponds to your operating system, click the link to download the ZIP file, and install the software.

- c. From <http://curl.haxx.se/docs/caextract.html>, download the `ca-bundle.pem` certificates bundle in the directory where you installed cURL.
- d. At the command prompt, go to the directory where you installed cURL, and set the cURL environment variable, `CURL_CA_BUNDLE`, to the location of the certificates bundle that you downloaded.

For example:

```
C:\curl> set CURL_CA_BUNDLE=ca-bundle.pem
```

2. Find out the REST endpoint URL for your service.
 - a. Sign in to Oracle Cloud My Services, using the My Services URL from the welcome email.
 - b. Click  near the upper left corner of the page.
 - c. In the menu that appears, expand **Services**, and click **Compute Classic**.
The Instances page of the Compute Classic web console is displayed.
 - d. Click **Site** near the top of the page, and select the site for which you want to find out the REST endpoint URL.
 - e. In the Site Selector dialog box that is displayed, note the URL in the **REST Endpoint** field.

3. Identify the two-part user name.

The user name that you must use for REST API calls to Compute Classic is a special two-part name in the format: `/Compute-serviceInstanceID/username`
username is the user name that you use to sign in to My Services.

To find out the service instance ID:

- a. Sign in to the Oracle Cloud My Services. The Oracle Cloud My Services Dashboard page is displayed. It lists the services available in your account.
- b. In the Compute service tile, click **Compute**. The service details page for Compute Classic is displayed.
- c. Under **Additional Information**, locate the **Service Instance ID** field and note the value displayed.

For example, if your service instance ID is 575260584 and the user name in your account-creation email is `jack.jones@example.com`, then use the following two-part user name for REST API calls:

```
/Compute-575260584/jack.jones@example.com
```

Get an Authentication Token

REST API requests to Compute Classic require basic authentication (user name and password). You can pass your user name and password with every API call or you can pass a valid authentication token.

To get an authentication token, send an request to authenticate the user credentials. If the authentication request succeeds, the server returns a cookie containing an authentication token that is valid for 30 minutes. The client sending the REST API requests must include this cookie in the API requests.

1. Get an authentication cookie from Compute Classic, as shown in the following cURL command example:

```
curl -i -X POST
  -H "Content-Type: application/oracle-compute-v3+json"
  -d '{"user":"/Compute-575260584/
jack.jones@example.com", "password": "ft7)Dvjo"}'
https://api-z999.compute.us0.oraclecloud.com/authenticate/
```

Enter the command on a single line. Line breaks are used in this example for readability.

- 575260584 and jack.jones@example.com are example values. Replace 575260584 with the service instance ID of your Compute Classic account, and jack.jones@example.com with your user name.
 - api-z999.compute.us0.oraclecloud.com is an example REST endpoint URL. Change this value to the REST endpoint URL of your Compute Classic site. For information about finding out REST endpoint URL for your site, see [Prepare to Use the REST API](#).
2. In the response to the POST request, look for the Set-Cookie header, as shown in the following example.

```
Set-Cookie:
nimbula=eyJpZGVudG10eSI6ICJ7XCJyZWZsbVwiOiBcImNvbXBldGUTdXM2LXoyOFwiLCBcIn
ZhbHVlXCI6IFwielxcXCJjdXN0b2llclxcXCi6IFxcXCJDb2lwdXRlLWFjbWVjY3NcXFwiLCBc
XFwicmVhbG1cXFwiOiBcXFwiY29tcHV0ZS1lczyteji4XFxcIiwgXFxcImVudG10eV90eXB1XF
xcIjogXFxcInVzZXJcXFwiLCBcXFwic2Vzc2l2b2l9leHBpcmVzXFxcIjogMTQ2MDQ4NjA5Mi44
MDM1NiwgXFxcImV4cGlyZXNcXFwiOiAxNDYwNDc3MDkyLjgWmZU5MiwgXFxcInVzZXJcXFwiOi
BcXFwiL0NvbXBldGUTYWNTZWNjcy9zeWxhamEua2FubmFuQG9yYWNsZS5jb2lcXFwiLCBcXFwi
Z3JvdXBzXFxcIjogW1xcXCiVQ29tcHV0ZS1hY211Y2NzL0NvbXBldGUuQ29tcHV0ZV9PcGVyYX
Rpb25zXFxcIiwgXFxcIi9Db2lwdXRlLWFjbWVjY3MvQ29tcHV0ZS5Db2lwdXRlX01vbm10b3Jc
XFwiXX1cIiwgXCJzaWduYXR1cmVcIjogXCJRT0xaeUZzdU54SmdjL3FuSk16MDRnNmRWNng2bl
Y5S0JpYm5zeFNCWXJxcVJVJGZmMkZtdjhoTytaVnZwQVdURGpwcZRNMHZTc2RocWw3QmM0VGJp
SmhFTWVyNFBjVGVvb05qd2VpaUcyaStBeDBPwmc3SDJFSjRITWQ0S1V3eTl6N1YzRhd4eUhwTj
dqM0w0eEFUTDUyeVpVQWVQK1diMkdzU1pjMmpTaHZyNi9ibU1CZ1Nyd2M4MUdxURBMFN6d044
V2VneUF1YVks5QTUxZmxaanJBMGVvVUJudmZ6NGxCUVVIZXl0Yyt0SXZVaDdUcGU2RGwxd3RSeF
NGVl1QR0FEQk9xMEExGaVd1QlpaU0FTZVcwOHBZcEZ2a2l0ZXdpdU9LaU93dFczV3VhFTZ3VHT0E1
YklibzYvMm50ZEhTWHJhYmYsY000UVE1LzZUMDJlZUpTYVE9PVwifSJ9; Path=/; Max-
Age=1800
```

Note that the Set-Cookie header name and value are in a single line. Line breaks are used in this example for readability.

3. Store the authentication cookie in an environment variable, as shown in the following example for a Linux host.

```
export
COMPUTE_COOKIE='nimbula=eyJpZGVudG10eSI6ICJ7XCJyZWZsbVwiOiBcImNvbXBldGUTdX
M2LXoyOFwiLCBcInZhbHVlXCI6IFwielxcXCJjdXN0b2llclxcXCi6IFxcXCJDb2lwdXRlLWFj
bWVjY3NcXFwiLCBcXFwicmVhbG1cXFwiOiBcXFwiY29tcHV0ZS1lczyteji4XFxcIiwgXFxcIm
VudG10eV90eXB1XFxcIjogXFxcInVzZXJcXFwiLCBcXFwic2Vzc2l2b2l9leHBpcmVzXFxcIjog
MTQ2MDQ4NjA5Mi44MDM1NiwgXFxcImV4cGlyZXNcXFwiOiAxNDYwNDc3MDkyLjgWmZU5MiwgXF
xcInVzZXJcXFwiOiBcXFwiL0NvbXBldGUTYWNTZWNjcy9zeWxhamEua2FubmFuQG9yYWNsZS5j
b2lcXFwiLCBcXFwiZ3JvdXBzXFxcIjogW1xcXCiVQ29tcHV0ZS1hY211Y2NzL0NvbXBldGUuQ2
```

```
9tcHV0ZV9PcGVyYXRpb25zXFxcIiwgXFxcIi9Db21wdXR1LWFjbWVjY3MvQ29tcHV0ZS
5Db21wdXR1X01vbml0b3JcXFwiXX1cIiwgXCJzaWduYXR1cmVcIjogXCJRT0xaeUZZdU
54SmdjL3FuSk16MDRnNmRWVng2blY5S0JpYm5zeFNCWXJXcVVJVGVZmMkZtdjhoTytaVn
ZwQVdURGpwcZRNMHZTc2RocWw3QmM0VGJpSmhFTWVyNFBjVVGvb05qd2VpaUcyStBeD
BPWmc3SDJFSjRITWQ0S1V3eTl6NlYzRHd4eUhwTjddM0w0eEFUTDUyeVpVQWVQK1diMk
dzU1pjMmpTaHZyNi9ibU1CZ1Nyd2M4MudxdURBMFN6d044V2VneUF1YVvk5QTUxZmxaan
JBMGVvVUJudmZ6NGxCUVVIZXloYyt0SXZVaDdUcGU2RGwxd3RSeFNGVV1QR0FEQk9xME
xGaVd1Q1paU0FTZVcwOHBZcEZ2a2l0ZXdpdU9LaU93dFc3VkFtZ3VHT0E1Yk1ibzYvMm
5oZEhTWHJhYmtsY000UVE1LzZUMDJlZUpTYVE9PVwifSJ9; Path=/; Max-
Age=1800'
```

Note that the `Set-Cookie` header and value are in a single line. Line breaks are used in this example for readability.

After getting an authentication cookie, you can perform operations on Compute Classic resources.

Get the Details of a VM Using the REST API

You can use the REST API to retrieve all the details of a VM.

1. To send REST API calls to Compute Classic, you need a valid authentication token. If you obtained a token less than 30 minutes ago, then you can use that token. Otherwise, get a new token as described in [Get an Authentication Token](#).
2. Send the following REST API request:

Syntax:

```
curl -X GET \
  -H "Cookie: $COMPUTE_COOKIE" \
  -H "Accept: application/oracle-compute-v3+json" \
  {restEndpointURL}/instance/{userName}/{instanceName}
```

If you don't know the REST endpoint URL and user name, follow the instructions in [Prepare to Use the REST API](#).

3. Example:

```
curl -X GET \
  -H "Cookie: $COMPUTE_COOKIE" \
  -H "Accept: application/oracle-compute-v3+json" \
  https://api-z999.compute.us0.oraclecloud.com/instance/
Compute-575260584/jack.jones@example.com/dev1/f653a677-
b566-4f92-8e93-71d47b364119
```

The response body contains all the details of the specified instance, in JSON format.

Add Block Storage for a VM Using the REST API

In this example, let's consider that you want to create a storage volume with the name `/Compute-acme/jack.jones@example.com/vol1` and attach it to an existing instance named `/Compute-acme/jack.jones@example.com/instance1`. After creating

the storage volume, you can create a storage attachment to attach the storage volume to the instance.

To provide block storage capacity for a Compute Classic instance, you must create one or more storage volumes and attach them to the instance.

1. Identify the REST endpoint URL and the two-part user name. If you don't know the REST endpoint URL and user name, follow the instructions in [Prepare to Use the REST API](#).

In this example, `https://api-z999.compute.us0.oraclecloud.com` and `/Compute-acme/jack.jones@example.com` are the example values for REST endpoint URL and the two-part user name respectively.

2. To send REST API calls to Compute Classic, you need a valid authentication token. If you obtained a token less than 30 minutes ago, then you can use that token. Otherwise, get a new token as described in [Get an Authentication Token](#).
3. Specify the details of the storage volume that you want to create in a JSON file.

The following shows an example of the request body content in the `storageVolume.json` file.

```
{
  "size": "10G",
  "properties": ["/oracle/public/storage/default"],
  "name": "/Compute-acme/jack.jones@example.com/vol1"
}
```

4. Create a storage volume by sending the `POST /storage/volume/` HTTP request.

Syntax

```
curl -X POST \
  -H "Cookie: $COMPUTE_COOKIE" \
  -H "Content-Type: application/oracle-compute-v3+json" \
  -H "Accept: application/oracle-compute-v3+json" \
  -d "@storageVolume.json" \
  {restEndpointURL}/storage/volume/
```

Example

```
curl -i -X POST \
  -H "Cookie: $COMPUTE_COOKIE" \
  -H "Content-Type: application/oracle-compute-v3+json" \
  -H "Accept: application/oracle-compute-v3+json" \
  -d "@storageVolume.json" \
  https://api-z999.compute.us0.oraclecloud.com/storage/volume/
```

Example of Response Body

The following example shows the response body in JSON format.

```
{
  "status": "Initializing",
  "account": "/Compute-acme/default",
  "writecache": false,
  "managed": true,
  "description": null,
  "tags": [],
  "bootable": false,
  "hypervisor": null,
  "quota": null,
  "uri": "https://api-z999.compute.us0.oraclecloud.com/storage/volume/Compute-acme/jack.jones@example.com/vol1",
}
```

```
"status_detail": "The storage volume is currently being initialized.",
"imagelist_entry": -1,
"storage_pool": "/Compute-acme/storagepool/iscsi/thruput_1",
"machineimage_name": null,
"status_timestamp": "2015-06-01T11:15:57Z",
"shared": false,
"imagelist": null,
"size": "10737418240",
"properties": ["/oracle/public/storage/default"],
"name": "/Compute-acme/jack.jones@example.com/voll"
}
```

- Retrieve details of the storage volume to check if the storage volume was created successfully by sending the GET /storage/volume/{name} HTTP request.

Syntax

```
curl -X GET \
  -H "Cookie: $COMPUTE_COOKIE" \
  -H "Accept: application/oracle-compute-v3+json" \
  {restEndpointURL}/storage/volume/{userName}/{storageVolumeName}
```

Example

```
curl -i -X GET \
  -H "Cookie: $COMPUTE_COOKIE" \
  -H "Accept: application/oracle-compute-v3+json" \
  https://api-z999.compute.us0.oraclecloud.com/storage/volume/Compute-acme/jack.jones@example.com/voll
```

Look at the value of the `status` field. When the storage volume is created, the value of the `status` field is `online`.

- Specify the details of the storage attachment that you want to create in a JSON file. You have to specify the name of the storage volume and the name of the instance to which you want to attach the storage volume.

Example of Request Body

The following shows an example of the request body content in the `storageAttach.json` file.

```
{
  "index": 1,
  "storage_volume_name": "/Compute-acme/jack.jones@example.com/voll",
  "instance_name": "/Compute-acme/jack.jones@example.com/instance1"
}
```

- Create a storage attachment to attach the storage volume to your instance by sending the POST /storage/attachment/ HTTP request.

Syntax

```
curl -X POST \
  -H "Cookie: $COMPUTE_COOKIE" \
  -H "Content-Type: application/oracle-compute-v3+json" \
  -H "Accept: application/oracle-compute-v3+json" \
  -d "@storageAttach.json" \
  {restEndpointURL}/storage/attachment/
```

Example

```
curl -i -X POST
  -H "Cookie: $COMPUTE_COOKIE"
```

```
-H "Content-Type: application/oracle-compute-v3+json"
-H "Accept: application/oracle-compute-v3+json"
-d "@storageAttach.json"
  https://api-z999.compute.us0.oraclecloud.com/storage/attachment/
```

Example of Response Body

The following example shows the response body in JSON format.

```
{
  "index": 1,
  "account": null,
  "storage_volume_name": "/Compute-acme/jack.jones@example.com/vol1",
  "hypervisor": null,
  "uri": "https://api-z999.compute.us0.oraclecloud.com/storage/attachment/Compute-acme/jack.jones@example.com/instance1/a7fb4550-df19-497c-a19f-44fc176e1fc2",
  "instance_name": "/Compute-acme/jack.jones@example.com/instance1",
  "state": "attaching",
  "readonly": false,
  "name": "/Compute-acme/jack.jones@example.com/instance1/a7fb4550-df19-497c-a19f-44fc176e1fc2"
}
```

In the response, note the name of the storage attachment.

8. Check whether the volume is attached by sending the GET `/storage/attachment/{name}` HTTP request.

Syntax

```
curl -X GET \
  -H "Cookie: $COMPUTE_COOKIE" \
  -H "Accept: application/oracle-compute-v3+json" \
  {restEndpointURL}/storage/volume/{userName}/{instanceName}/
{storageAttachmentName}
```

Example

```
curl -i -X GET \
  -H "Cookie: $COMPUTE_COOKIE" \
  -H "Accept: application/oracle-compute-v3+json" \
  https://api-z999.compute.us0.oraclecloud.com/storage/attachment/Compute-acme/
jack.jones@example.com/instance1/a7fb4550-df19-497c-a19f-44fc176e1fc2
```

Look at the value of the `status` field. After the volume is attached, the `status` field shows `attached`.

9. Mount and format the disk that you just attached. See [Mounting and Unmounting a Storage Volume in *Using Oracle Cloud Infrastructure Compute Classic*](#).



See Also:

About Storage Volumes in *Using Oracle Cloud Infrastructure Compute Classic*.

12

Object Storage Classic: Managing Containers and Objects

Topics

- [Get an Authentication Token](#)
- [Create a Container](#)
- [List the Containers in the Account](#)
- [Upload a Large File](#)
- [Download a File](#)
- [Copy an Object](#)
- [List the Objects in a Container](#)
- [Delete an Object](#)
- [Delete a Container](#)

Get an Authentication Token

When you send REST API requests to Oracle Cloud Infrastructure Object Storage Classic, you must include an authentication token. You request an authentication token by sending your user credentials to the service. Authentication tokens are temporary; they expire after 30 minutes.

1. Send a `GET` request to the authentication endpoint URL.

Syntax

```
curl -i -X GET \  
  {authenticationEndpointURL} \  
  -H 'x-storage-user: Storage-{accountName}:{userName}' \  
  -H 'x-storage-pass: {password}'
```

- You can find out the authentication endpoint URL from the service details page of Oracle Cloud My Services.
- `accountName` is the account name.
- `userName` and `password` are the credentials you use to sign in to Oracle Cloud My Services.

Sample Command

```
curl -i -X GET \  
  https://myaccount.ocm.rack01.example.com/auth/v1.0 \  
  \
```

```
-H 'x-storage-user: Storage-myaccount:myusername' \  
-H 'x-storage-pass: mypassword'
```

2. In the response, look for the **X-Auth-Token** header and note its value.

Sample Response Headers

```
HTTP/1.1 200 OK  
Server: nginx/1.10.2  
Date: Wed, 01 Aug 2018 01:05:44 GMT  
Content-Length: 0  
Connection: close  
X-Auth-Token: AUTH_tk10d7cf10041726fa2e64652d975bbab0  
X-Storage-Token: AUTH_tk10d7cf10041726fa2e64652d975bbab0  
X-Storage-Url: https://myaccount.ocm.rack01.example.com/v1/Storage-  
ids-63b8bbbbbb64085920856f814f06720
```

Create a Container

A container is a storage compartment that provides a way to organize the data that's stored in Oracle Cloud Infrastructure Object Storage Classic.

Before you send REST API calls to Oracle Cloud Infrastructure Object Storage Classic, you need a valid authentication token. If you obtained a token less than 30 minutes ago, then you can use that token. Otherwise, to get a new token, see [Get an Authentication Token](#).

1. Send a **PUT** request, specifying the name of the container. Specify the authentication token in the **x-auth-token** header.

Syntax

```
curl -i -X PUT \  
  {accountRestEndpointURL}/{containerName} \  
  -H 'x-auth-token: {authToken}'
```

Sample Command

```
curl -i -X PUT \  
  https://myaccount.ocm.rack01.example.com/v1/Storage-myaccount/  
mycontainer \  
  -H 'x-auth-token: AUTH_tk10d7cf10041726fa2e64652d975bbab0'
```

2. In the response, look for 201 Created.

```
HTTP/1.1 201 Created  
Server: nginx/1.10.2  
Date: Thu, 02 Aug 2018 22:46:23 GMT  
Content-Length: 0  
Connection: close  
X-Last-Modified-Timestamp: 1533249983.19669  
X-Trans-Id: txc19a77366846429a93634-005b6389bfga
```

List the Containers in the Account

To list the containers in an account, send a `GET` request to the account.

1. To send REST API calls to Object Storage Classic, you need a valid authentication token. If you obtained a token less than 30 minutes ago, then you can use that token. Otherwise, get a new token as described in [Get an Authentication Token](#).
2. Send a `GET` call to the account. Specify the authentication token in the **X-Auth-Token** header.

Syntax

```
curl -X GET \  
  {accountRestEndpointURL} \  
  -H 'x-auth-token: {authToken}'
```

Sample Command

```
curl -X GET \  
  https://myaccount.ocm.rack01.example.com/v1/Storage-myaccount \  
  -H 'x-auth-token: AUTH_tk10d7cf10041726fa2e64652d975bbab0'
```

3. The output is a list of the containers in the account.

Upload a Large File

A file that's larger than 5 GB is considered a large object. A single object can hold up to 5 GB of data, but multiple objects can be linked together to hold more than 5 GB of contiguous data. You can create small objects as segments and upload them as one large object by using a manifest object.



Note:

A large object can have a maximum of 2048 segments. Each segment can be up to 5 GB. The maximum size of a file that you can upload to Oracle Cloud Infrastructure Object Storage Classic as a large object is 10 TB.

A user with the Service Administrator role or a role that is specified in the X-Container-Write ACL of the container can perform this task.

You can upload a large object by using the REST API.

1. Segment the large file locally into multiple sequential segment files, each smaller than 5 GB.

On Linux, for example, you can use the following command:

```
split -b segment_size file_name segment_name
```

2. List all the segment files.

```
ls -al segment_name*
```

3. Create objects from each segment file. Upload all the objects in the same container.

```
curl -v -X PUT \
  -H "X-Auth-Token:token" \
  -T segmentName \
  accountURL/containerName/objectName
```

- **token:** The authentication token obtained from Oracle Cloud Infrastructure Object Storage Classic
 - **segmentName:** The full path and name of the segment file to be uploaded
 - **containerName:** The name of the container in which the object should be created
 - **objectName:** The name of the object to be created, which is same as the corresponding segment file name
4. Create a manifest file in JSON format, and ensure that the manifest file contains the following attributes for each segment object:
- **path:** The container and object name in the format:
containerName/segmentObjectName
 - **etag:** MD5 checksum of the segment object.
You can find the value in the `Etag` header of the segment object.
 - **size_bytes:** Size of the segment object.
You can find the value in the `Content-Length` header of the segment object.

Sample Manifest File

```
[
  {
    "path": "FirstContainer/segment_aa",
    "etag": "f1c9645dbc14efddc7d8a322685f26eb",
    "size_bytes": 10485760
  },
  {
    "path": "FirstContainer/segment_ab",
    "etag": "f1c9645dbc14efddc7d8a322685f26eb",
    "size_bytes": 10485760
  },
  {
    "path": "FirstContainer/segment_ac",
    "etag": "f1c9645dbc14efddc7d8a322685f26eb",
    "size_bytes": 10485760
  },
  {
    "path": "FirstContainer/segment_ad",
    "etag": "f1c9645dbc14efddc7d8a322685f26eb",
    "size_bytes": 10485760
  },
  ...
  {
    "path": "FirstContainer/segment_aj",
    "etag": "f1c9645dbc14efddc7d8a322685f26eb",
    "size_bytes": 10485760
  }
]
```

5. Upload the manifest file that you just created. In the URI, include the `?multipart-manifest=put` query parameter.

```
curl -v -X PUT \
  -H "X-Auth-Token:token" \
  "accountURL/containerName/LargeFileName?multipart-manifest=put" \
  -T ./fileName.json
```

- LargeFileName: The name of the large object
- fileName.json: The name of the manifest file
- ?multipart-manifest=put: The query parameter to upload the manifest file

6. Check the size of the large object.

```
curl -v -X HEAD \
  -H "X-Auth-Token:token" \
  accountURL/containerName/LargeObjectName
```

The size of the large object is the total size of all the segment objects.

Example:

The following example shows how to upload a large file, using an Oracle Cloud account with the following details:

- Account name: acme
- REST Endpoint URL: <https://acme.storage.oraclecloud.com/v1/Storage-acme>
- REST Endpoint (Permanent) URL: <https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365>

Note:

The REST Endpoint (Permanent) URL is displayed for the accounts created after November 2017.

The example uses the REST Endpoint URL for the sample Oracle Cloud account. To use the REST Endpoint (Permanent) URL, replace <https://acme.storage.oraclecloud.com/v1/Storage-acme> with <https://storage-7b16fede61e1417ab83eb52e06f0e365.storage.oraclecloud.com/v1/Storage-7b16fede61e1417ab83eb52e06f0e365>.

1. Segment the large file locally into multiple sequential segment files, each smaller than 5 GB:

```
split -b 10m myLargeFile.zip segment_
```

2. List all the segment files:

```
ls -al segment_*
```

```
segment_aa
segment_ab
segment_ac
segment_ad
segment_ae
segment_af
segment_ag
segment_ah
segment_ai
segment aj
```


3. Create objects from each segment file (*segment_aa*, *segment_ab*...*segment_aj*), preserving the segment file names. Upload all the objects in the `FirstContainer` container. The following example shows one of the segment files:

```
curl -v -X PUT \
  -H "X-Auth-Token: AUTH_tk5a58b7a8c34bb7b662523a59a5272650" \
  -T segment_aa
  https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/
segment_aa
```

The following example shows the output of this command:

```
> User-Agent: curl/7.29.0
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkc9305a46ebaa0585c4c7ae063c844f0b
> Content-Length: 10485760
> Expect: 100-continue
>
< HTTP/1.1 100 Continue
* We are completely uploaded and fine
< HTTP/1.1 201 Created
< Date: Tue, 15 Dec 2015 10:18:26 GMT
< Last-Modified: Tue, 15 Dec 2015 10:17:21 GMT
< X-Trans-Id: tx85da332ec5ae4852b7d8c-00566fe8b0ga
< Etag: f1c9645dbc14efddc7d8a322685f26eb
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1450174640.10123
< Content-Type: text/html;charset=UTF-8
< Content-Length: 0
<
```

4. Create a manifest file in JSON format.

Example Manifest File

```
[
  {
    "path": "FirstContainer/segment_aa",
    "etag": "f1c9645dbc14efddc7d8a322685f26eb",
    "size_bytes": 10485760
  },
  {
    "path": "FirstContainer/segment_ab",
    "etag": "f1c9645dbc14efddc7d8a322685f26eb",
    "size_bytes": 10485760
  },
  {
    "path": "FirstContainer/segment_ac",
    "etag": "f1c9645dbc14efddc7d8a322685f26eb",
    "size_bytes": 10485760
  },
  {
    "path": "FirstContainer/segment_ad",
    "etag": "f1c9645dbc14efddc7d8a322685f26eb",
    "size_bytes": 10485760
  },
  ...
  {
    "path": "FirstContainer/segment_aj",
    "etag": "f1c9645dbc14efddc7d8a322685f26eb",
```

```

        "size_bytes": 10485760
    }
]

```

5. Upload the manifest file. Add the `?multipart-manifest=put` query parameter to upload the manifest file.

```

curl -v -X PUT \
  -H "X-Auth-Token: AUTH_tk5a58b7a8c34bb7b662523a59a5272650"
  "https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/
myLargeFile.manifest?multipart-manifest=put" \
  -T ./manifest.json

```

6. Download the large object by sending a `GET` request. All the segment objects are concatenated and downloaded as one large object.

```

curl -v -X GET \
  -H "X-Auth-Token: AUTH_tk5a58b7a8c34bb7b662523a59a5272650"
  https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/
myLargeFile.manifest \
  -o ./myLargeFile

```

7. Download the manifest object by sending a `GET` request, and add the `?multipart-manifest=get` query parameter.

```

curl -v -X GET \
  -H "X-Auth-Token: AUTH_tk5a58b7a8c34bb7b662523a59a5272650"
  "https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/
myLargeFile.manifest?multipart-manifest=get" \
  -o ./manifestFile

```

8. Run a `HEAD` request to view the size of the large object (`myLargeFile`) that you created:

```

curl -v -X HEAD \
  -H "X-Auth-Token: AUTH_tkbaebb60dfa5b80d84e62b0d5d07031e5"
  https://acme.storage.oraclecloud.com/v1/Storage-acme/FirstContainer/
myLargeFile

```

The following example shows the output of this command:

```

> HEAD /v1/Storage-acme/FirstContainer/myLargeFile HTTP/1.1
> User-Agent: curl/7.29.0
> Host: acme.storage.oraclecloud.com
> Accept: */*
> X-Auth-Token: AUTH_tkc9305a46ebaa0585c4c7ae063c844f0b
< Etag: "e6da53c20abee5c471fe8bf796abb1a4"
< Accept-Ranges: bytes
< Last-Modified: Tue, 15 Dec 2015 10:07:53 GMT
< X-Timestamp: 1455012472.56679
< X-Trans-Id: txcab964b91ba8474ca9193-0056b9bb6fga
< Date: Tue, 15 Dec 2015 10:12:00 GMT
< Connection: keep-alive
< X-Last-Modified-Timestamp: 1455012472.56679
< Content-Type: application/octet-stream; charset=UTF-8
< Content-Length: 104857600
curl: (18) transfer closed with 52428800 bytes remaining to read

```

You can view the size of the large object in the `Content-Length` header. The size of the large object is the sum total of the sizes of the segment objects.

Download a File

To download a file, send a `GET` request to the object.

1. To send REST API calls to Object Storage Classic, you need a valid authentication token. If you obtained a token less than 30 minutes ago, then you can use that token. Otherwise, get a new token as described in [Get an Authentication Token](#).
2. Send a `GET` request to the object. Specify the authentication token in the **x-auth-token** header.

Syntax

```
curl -X GET \  
  {accountRestEndpointURL}/{containerName}/{objectName} \  
  -H 'x-auth-token: {authToken}' \  
  -o {targetLocalFileName}
```

Sample Command

```
curl -X GET \  
  https://myaccount.ocm.rack01.example.com/v1/Storage-myaccount/  
mycontainer/myobject \  
  -H 'x-auth-token: AUTH_tk10d7cf10041726fa2e64652d975bbab0' \  
  -o myfile
```

Copy an Object

An object can be copied to another object within the same or another container. There is no need to download the object and then upload it again; the copying operation is performed entirely on the server.

1. To send REST API calls to Object Storage Classic, you need a valid authentication token. If you obtained a token less than 30 minutes ago, then you can use that token. Otherwise, get a new token as described in [Get an Authentication Token](#).
2. Send a `COPY` call to the object you want to copy, and specify the destination container and object in the `Destination` header.

Syntax

```
curl -X COPY \  
  {accountRestEndpointURL}/{sourceContainerName}/{sourceObjectName} \  
 \  
  -H 'Destination: {destinationContainerName}/  
{destinationObjectName}' \  
  -H 'x-auth-token: {authToken}'
```

Sample Command

```
curl -X COPY \  
  https://myaccount.ocm.rack01.example.com/v1/Storage-myaccount/  
mycontainer/myobject \  
  -H 'Destination: mycontainer/myobject'
```

```
-H 'Destination: myothercontainer/myobjectopy'  
-H 'x-auth-token: AUTH_tk10d7cf10041726fa2e64652d975bbab0'
```

3. The output is a list of the containers in the account.

List the Objects in a Container

To list the objects in a container, send a `GET` request to the container.

1. To send REST API calls to Object Storage Classic, you need a valid authentication token. If you obtained a token less than 30 minutes ago, then you can use that token. Otherwise, get a new token as described in [Get an Authentication Token](#).
2. Send a `GET` request to the container. Specify the authentication token in the **x-auth-token** header.

Syntax

```
curl -X GET \  
  {accountRestEndpointURL}/{containerName} \  
  -H 'x-auth-token: {authToken}'
```

Sample Command

```
curl -X GET \  
  https://myaccount.ocm.rack01.example.com/v1/Storage-myaccount/  
mycontainer \  
  -H 'x-auth-token: AUTH_tk10d7cf10041726fa2e64652d975bbab0'
```

3. The output is a list of the objects in the container.

Delete an Object

To remove an object permanently from a container, send a `DELETE` request to the object.

1. To send REST API calls to Object Storage Classic, you need a valid authentication token. If you obtained a token less than 30 minutes ago, then you can use that token. Otherwise, get a new token as described in [Get an Authentication Token](#).
2. Send a `DELETE` request to the object. Specify the authentication token in the **x-auth-token** header.

Syntax

```
curl -X DELETE \  
  {accountRestEndpointURL}/{containerName}/{objectName} \  
  -H 'x-auth-token: {authToken}'
```

Sample Command

```
curl -X DELETE \  
  https://myaccount.ocm.rack01.example.com/v1/Storage-myaccount/  
mycontainer/myobject \  
  -H 'x-auth-token: AUTH_tk10d7cf10041726fa2e64652d975bbab0'
```

3. In the response, look for **204 No Content**.

```
HTTP/1.1 204 No Content
Server: nginx/1.10.2
Date: Thu, 02 Aug 2018 23:05:55 GMT
Content-Type: text/plain;charset=UTF-8
Connection: close
X-Timestamp: 1533169820576
X-Last-Modified-Timestamp: 1533251155.84714
X-Trans-Id: tx803bcd4e5535450b93f79-005b638e53g
```

Delete a Container

When you no longer need a container, you can delete it.

1. Make sure that the container you want to delete is empty. If it isn't empty, then first delete the objects in it. See [Delete an Object](#).
2. To send REST API calls to Object Storage Classic, you need a valid authentication token. If you obtained a token less than 30 minutes ago, then you can use that token. Otherwise, get a new token as described in [Get an Authentication Token](#).
3. Send a `DELETE` request to the container. Specify the authentication token in the `x-auth-token` header.

Syntax

```
curl -X DELETE \
  {accountRestEndpointURL}/{containerName} \
  -H 'x-auth-token: {authToken}'
```

Sample Command

```
curl -X DELETE \
  https://myaccount.ocm.rack01.example.com/v1/Storage-myaccount/
mycontainer \
  -H 'x-auth-token: AUTH_tk10d7cf10041726fa2e64652d975bbab0'
```

4. In the response, look for **204 No Content**.

```
HTTP/1.1 204 No Content
Server: nginx/1.10.2
Date: Thu, 02 Aug 2018 23:11:08 GMT
Connection: close
X-Last-Modified-Timestamp: 1533251468.65728
X-Trans-Id: txb7dca7e05d7443dba0adb-005b638f8cg
```

A

Additional Cloud at Customer Tasks

When you are getting started, there are some additional Cloud at Customer account management tasks you often have to perform.

Topics:

- [Web Browser Requirements](#)
- [Change Your Cloud Account Password](#)

Web Browser Requirements

The following table lists supported browsers for using Oracle Cloud services. Some Oracle Cloud services and tools have additional or specific browser requirements. See the documentation for the Cloud Services you are using.

Web / Mobile Browser	Version
Microsoft Internet Explorer	11
Mozilla Firefox	Extended Support Release (ESR) 52 and later
Google Chrome	63 and later
Apple Safari	10 and later
Safari, Chrome, Firefox on iOS (iPad and iPhone)	Latest
Chrome, Firefox on Android (Phone and Tablet)	Latest

Change Your Cloud Account Password

You can change your password on the My Profile page. Note that passwords are valid only for a specified period as defined by your password policy.

To change your password:

1. Sign in to My Services.
2. Click your user name (typically your email address) at the top of the screen to display the user name menu.
3. From the Profile menu, select **My Profile**.
4. Click the **Change Password** tab.
5. Enter a new password in the **New Password** field.
For password requirements, review the **Password Criteria** pane and ensure that you meet the specified criteria.
6. Reenter your new password in the **Confirm New Password** field.
7. Click **Save**.

If successful, then you'll receive an email notification. See also Changing Your Password in *Administering Oracle Identity Cloud Service*.

Tip: In some cases, when you select **My Profile** from the user name menu, an incorrect page appears. If you don't see the My Profile page, which includes your Profile Details, Change My Password, and Set Email Options tabs, then do the following:

1. Note the URL that appears in the address field of your Web browser.

If you experience this issue, then the URL will likely appear as follows:

```
https://idcs-guid.identity.hostname.com/ui/v1/myconsole
```

2. Add the following additional string to the end of the URL:

```
/?root=my-info
```

After you edit the URL, it should appear as follows:

```
https://idcs-guid.identity.hostname.com/ui/v1/myconsole/?root=my-info
```

3. Use your browser to navigate to the newly edited URL.

You should see your profile settings, including the Change My Password tab.